# What can POP do for you?

Mike Dewar, NAG Ltd

EU H2020 Center of Excellence (CoE)

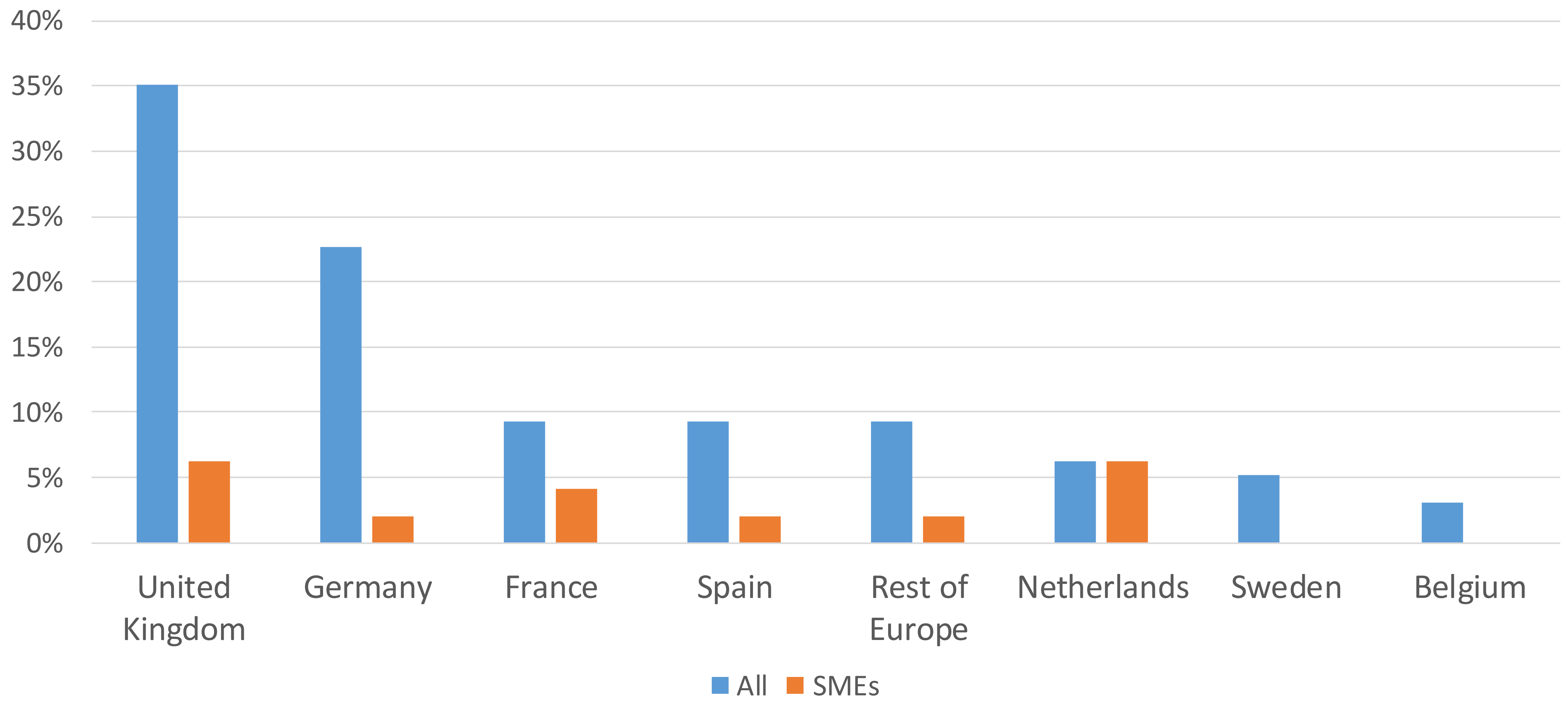
1 October 2015 – 31 March 2018

Grant Agreement No 676553

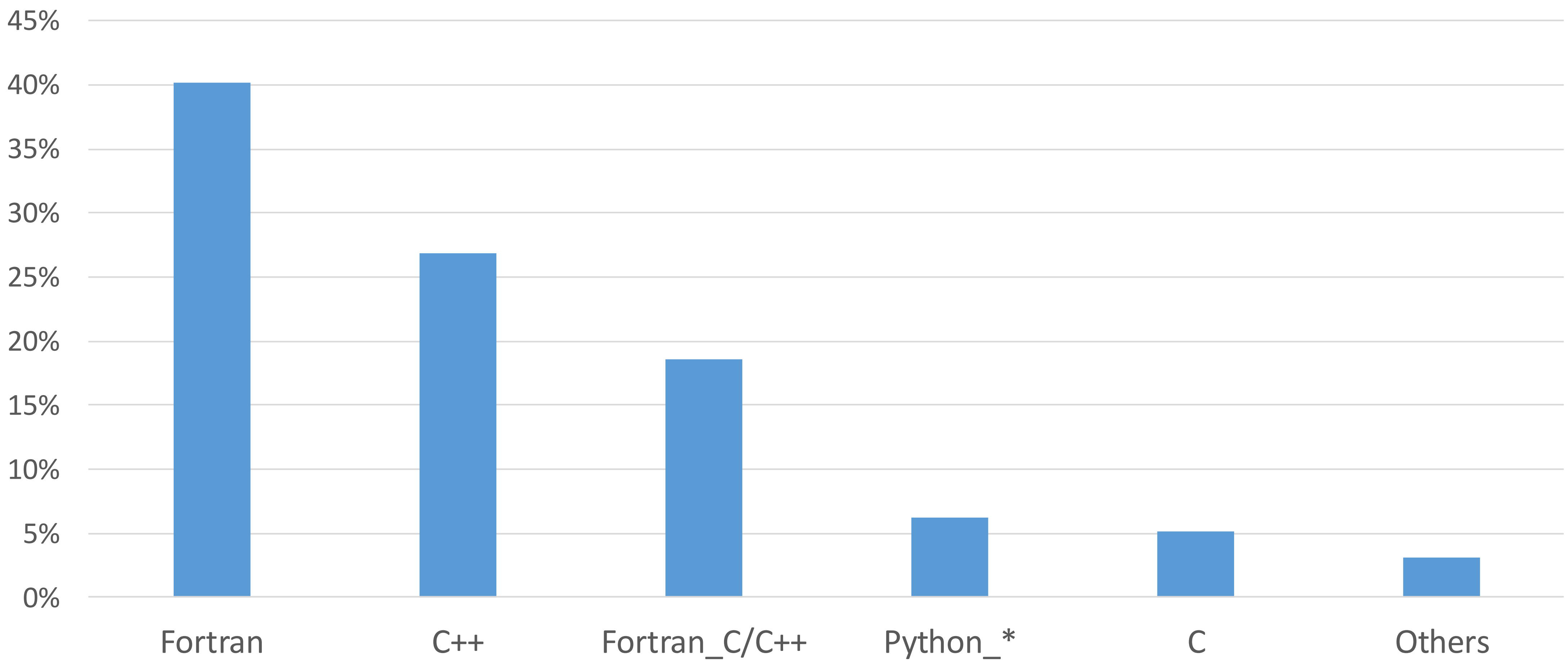
# Outline

- Overview of codes investigated
- Code audit & plan examples
- Analysis of inefficiencies identified
- Proof of concept projects
- Summary

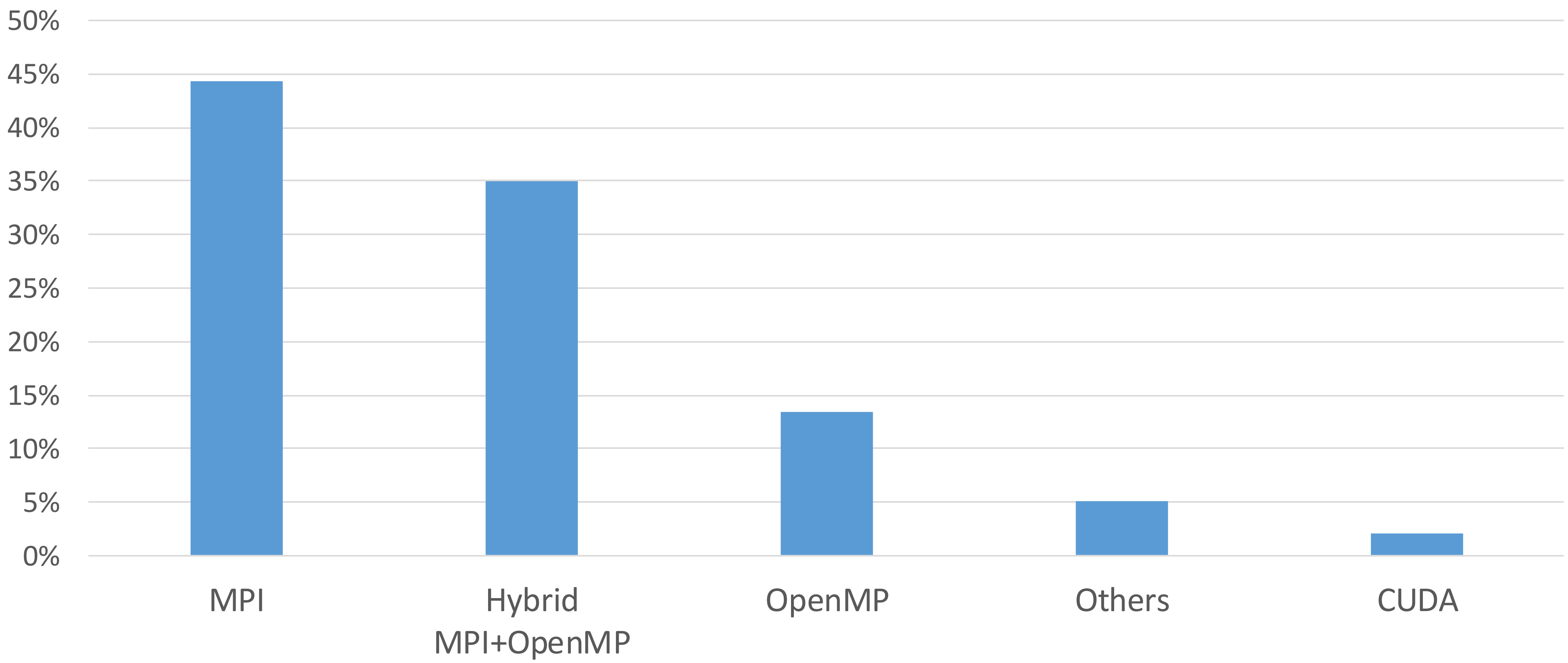
# Customers by Country



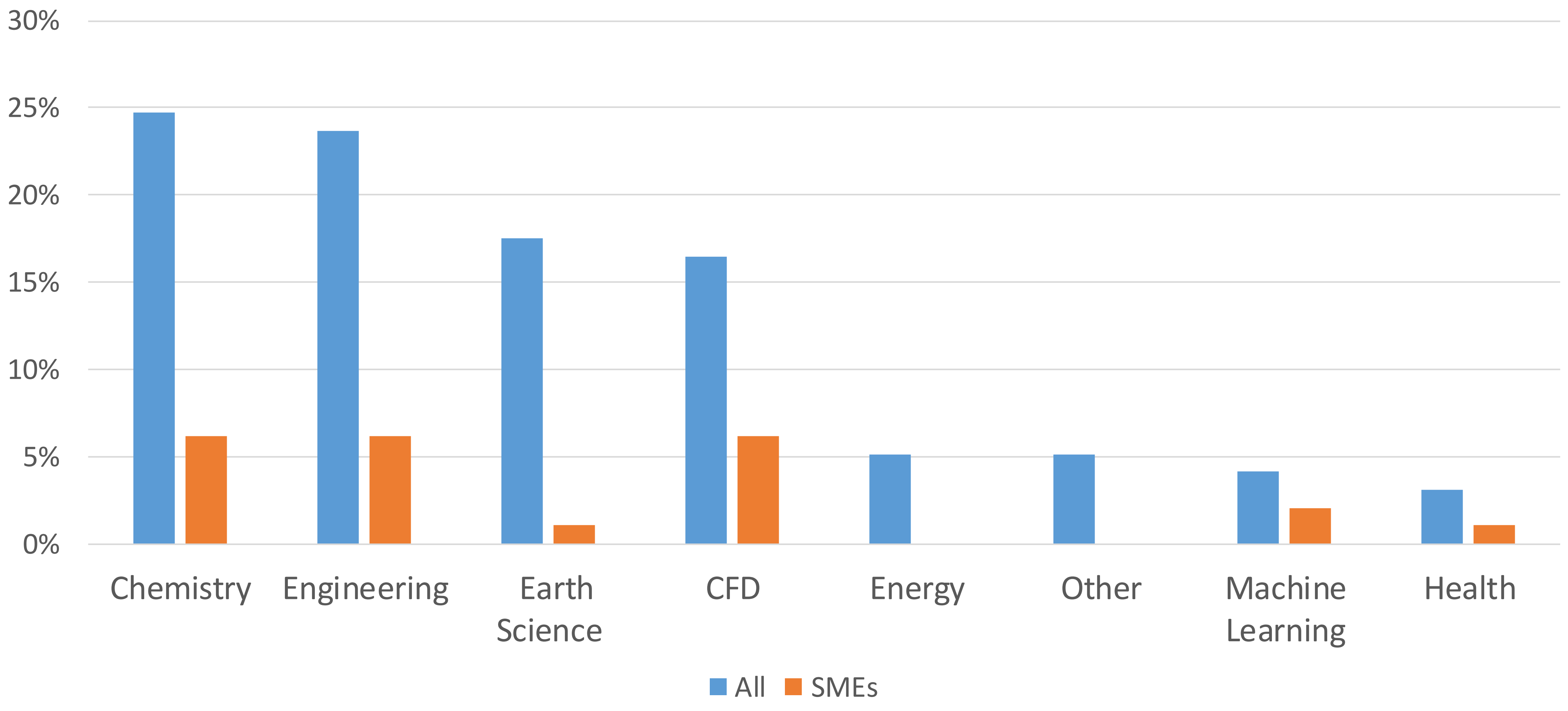
# Programming Languages



# Parallelisation Scheme



# Application Sectors



# So Far

- 72 Audits and plans completed or reporting to customer
- 5 completed Proofs of Concept
- Working on a further 36 studies and 8 Proofs of Concept
- Close to 40% of Audits lead to a follow-up Performance Plan

- Goal – 150 assessments

# Code Audit & Plan Examples

# OpenNN - Artelnics

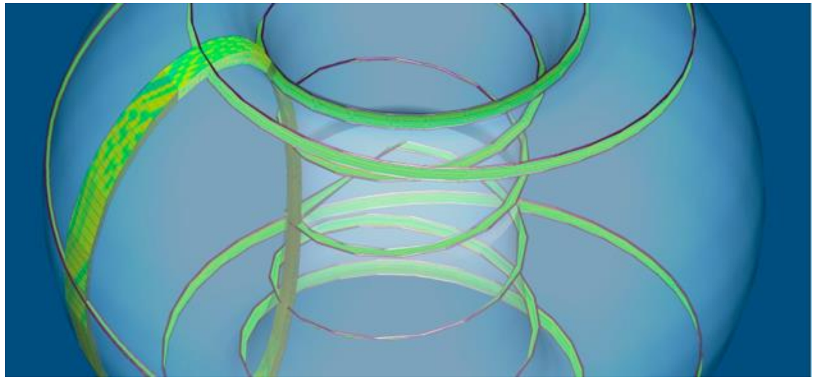
www.artelnics.com

- Neural network open source application
- C++ code with OpenMP parallelisation

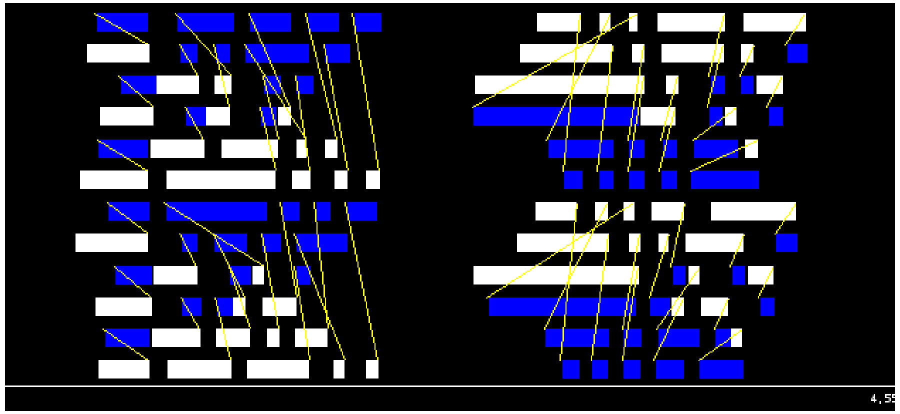
- Key audit result:
  - Main issue is Computational Efficiency
    - Main limit on performance is the unexpected variability in the number of times a parallel loop gets executed when number of threads is increased.
  - Further work in a POP performance plan is investigating the unexpected extra computation

# GS2 - Culham Centre for Fusion Energy

- Turbulence in fusion plasma application
- Fortran code with MPI parallelisation

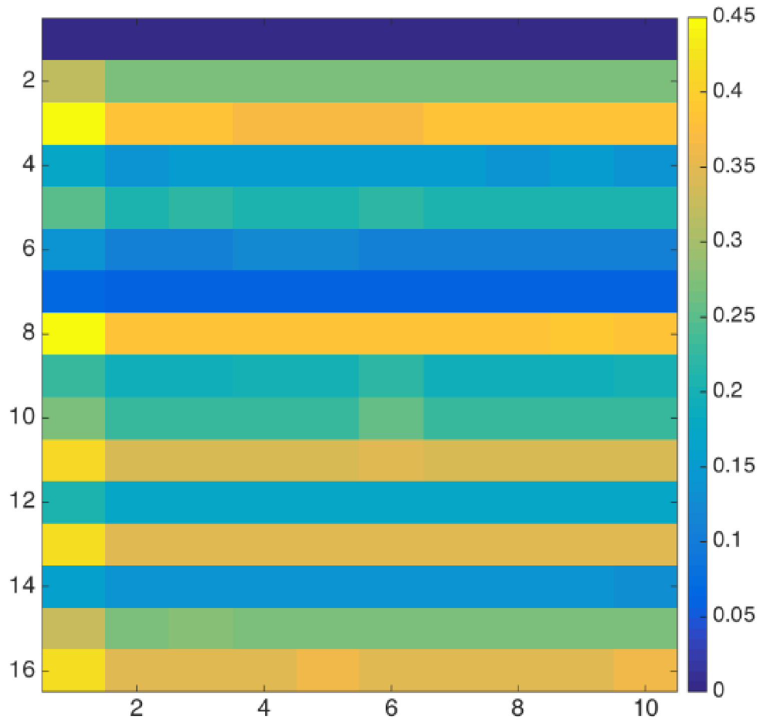


- Key audit result:
  - Main issue is Communication Efficiency
  - Serialisation in the point-to-point calls leading to waiting time
  - Use of non-blocking calls recommended



# DROPS – RWTH Aachen

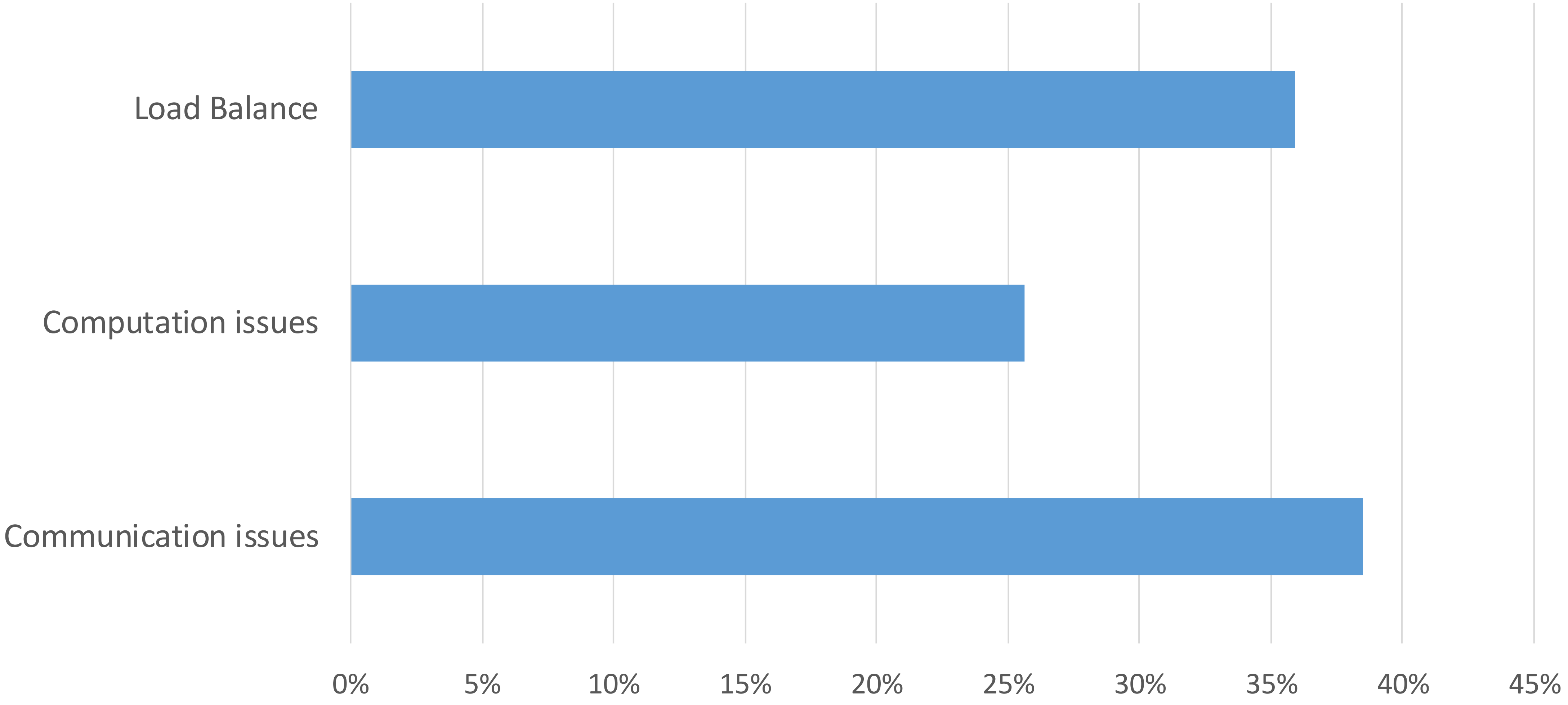
- CFD tool for simulating two-phase flows
- C++ code parallelised with Hybrid MPI + OpenMP
- Complex due to heavy use of C++ templates



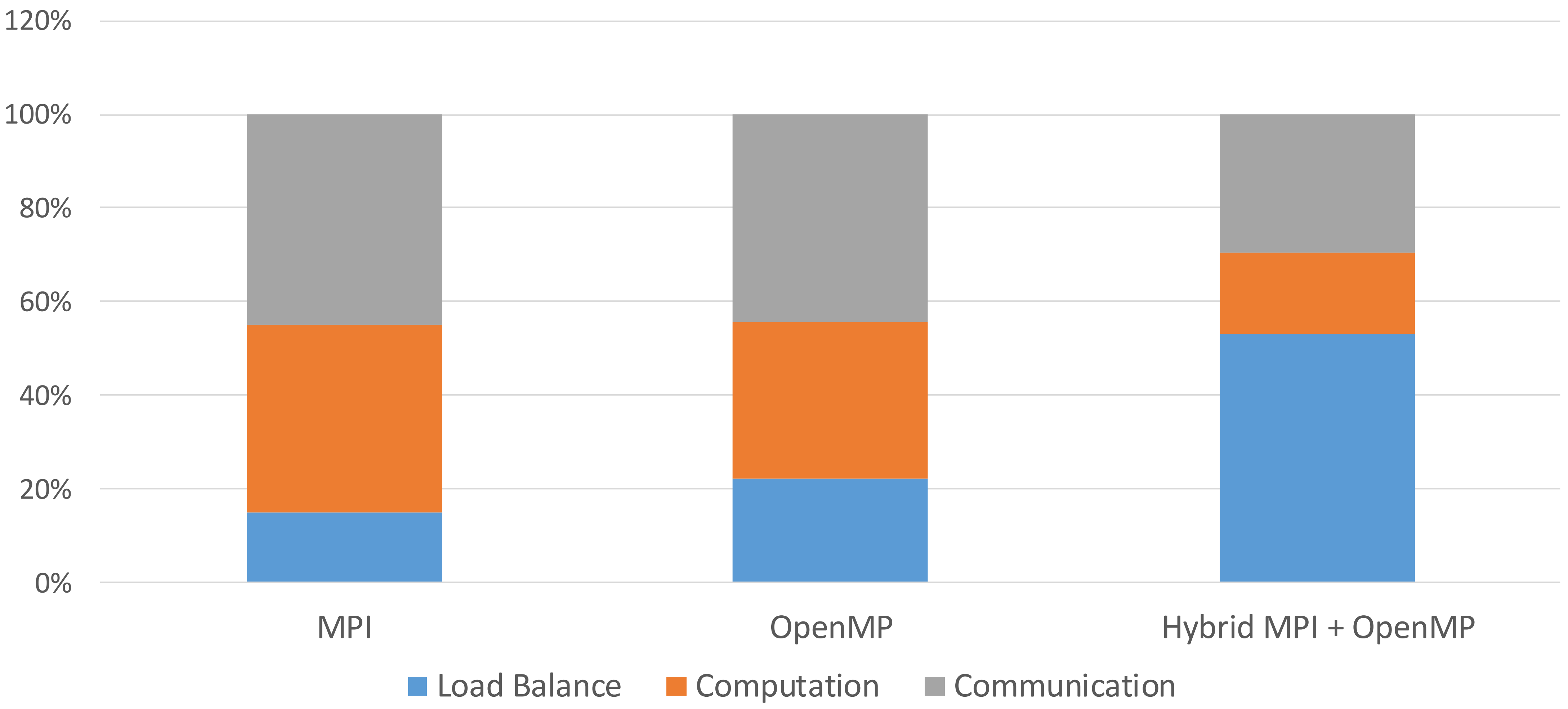

- Key audit result
  - Main issue with computational Load Balance
  - Resulted in waiting times in MPI collectives

# Analysis of Inefficiencies

# Leading cause of inefficiency



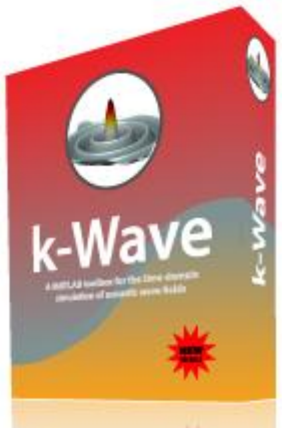
# Inefficiency by Parallelisation



# Proof of concept projects

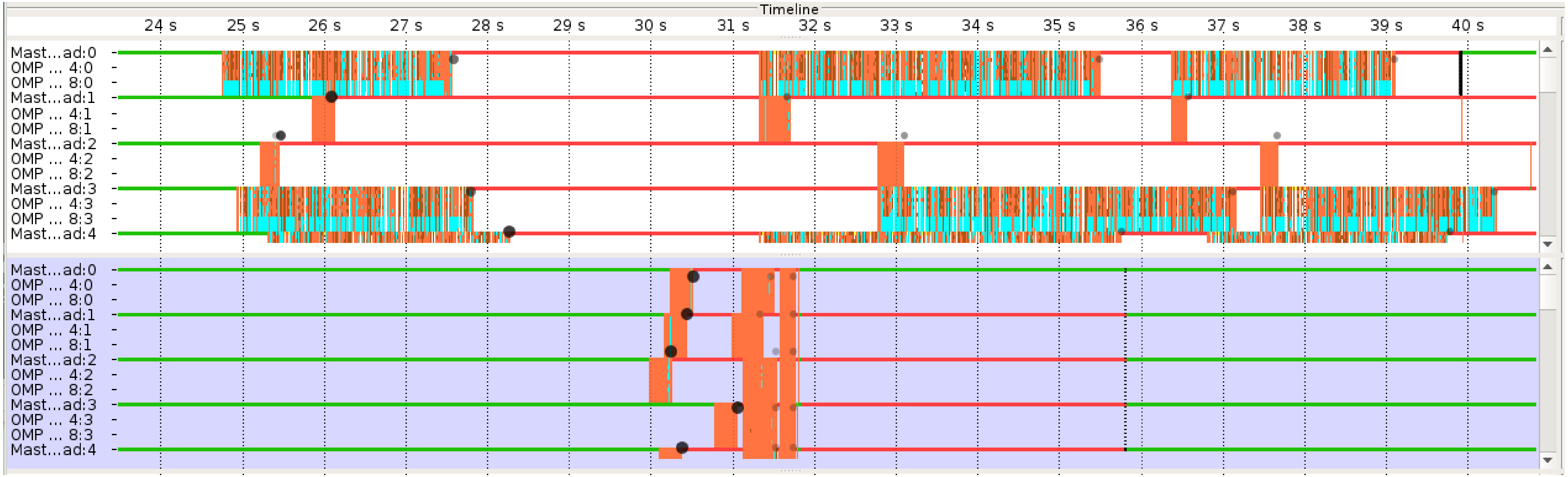
# k-Wave – Brno University of Technology

- Toolbox for time domain acoustic and ultrasound simulations in complex and tissue-realistic media
- C++ code parallelised with Hybrid MPI and OpenMP (+ CUDA)
- Executed on Salomon Intel Xeon compute nodes
- Key audit findings:
  - 3D domain decomposition suffered from major load imbalance : exterior MPI processes with fewer grid cells took much longer than interior
  - OpenMP-parallelised FFTs were much less efficient for grid sizes of exterior, requiring many more small and poorly-balanced parallel loops
- Using a periodic domain with identical halo zones for each MPI rank reduced overall runtime by a factor of 2




www.k-wave.org

# k-Wave – Brno University of Technology

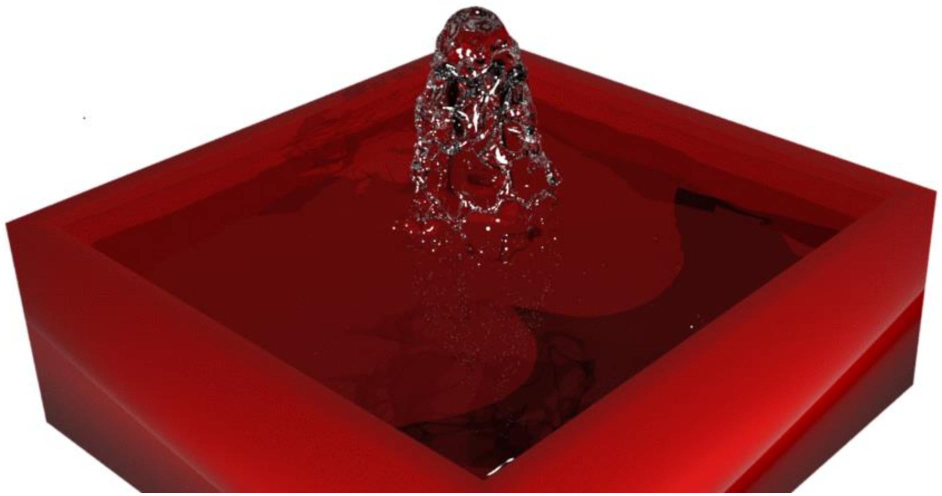



- Comparison time-line before (white) and after (lilac) balancing, showing exterior MPI ranks (0,3) and interior MPI ranks (1,2)
  - MPI synchronization in red, OpenMP synchronization in cyan

# sphFluids – Stuttgart Media University

- Simulates fluids for computer graphics applications
- C++ parallelised with OpenMP

- Key audit results:
  - Several issues relating to the sequential computational performance
  - Located critical parts of the application with specific recommended improvements



# sphFluids – Stuttgart Media University

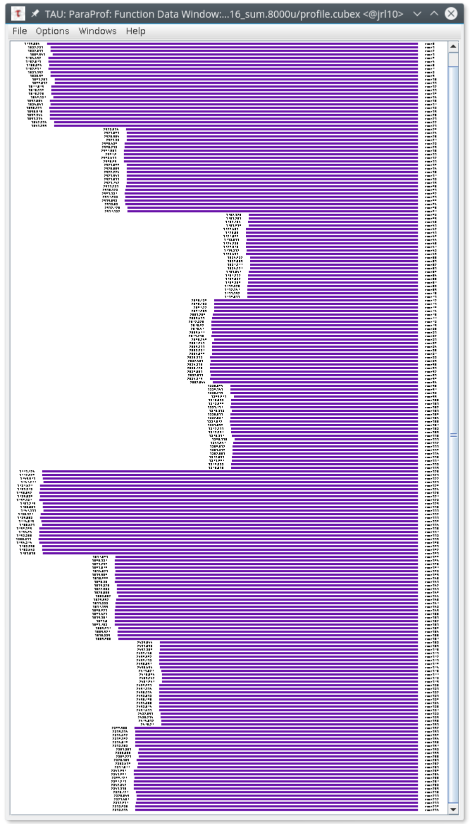
- Implemented by the code developers:
  - Review of overall code design from issues identified in POP audit
  - Inlining short functions
  - Reordering the particle processing order to reduce cache misses
  - Removal of unnecessary operations and costly inner loop definitions

- Confirmed performance improvement up to 5x – 6x depending on scenario and pressure model used
- Achieved thanks to insights provided by the POP experts and good information exchange during the work

# EPW – University of Oxford

- Electron-Phonon Wannier (EPW) materials science DFT code;
- Part of the Quantum ESPRESSO suite
- Fortran code parallelised with MPI
- Audit of unreleased development version of code
- Executed on ARCHER Cray XC30 (24 MPI ranks per node)

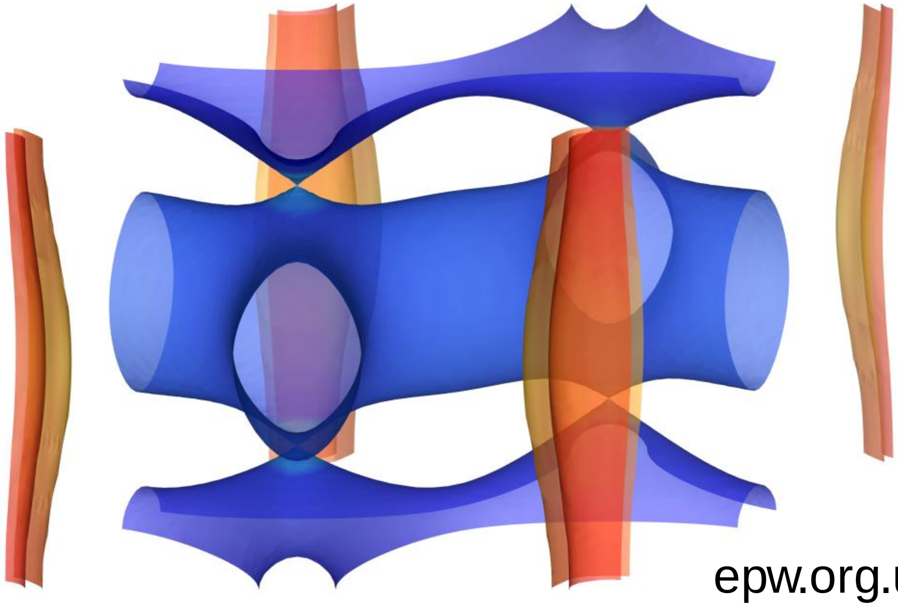
- Key audit findings:
  - Poor load balance from excessive computation identified (addressed in separate POP Performance Plan)
  - Large variations in runtime, likely caused by IO
  - Final stage spends a great deal of time writing output to disk



# EPW – University of Oxford

- Original code had all MPI ranks writing the result to disk at the end
- POP PoC modified this to have only one rank do output
- On 480 MPI ranks, time taken to write results fell from over 7 hours to 56 seconds: 450-fold speed-up!

- Combined with previous improvements, enabled EPW simulations to scale to previously impractical 1920 MPI ranks
- 86% global efficiency with 960 MPI ranks



epw.org.uk

# Summary

- POP seeks to not only describe the performance of an application, but to identify the root causes of poor performance.
- Better performance leads to both resource savings and improved science.
- POP is a free service for people and organisations in the European Union.
- Current funding secured until March 2018 – apply now for full range of services

https://pop-coe.eu

“POP analysis elegantly reveals in detail how our application's algorithm is running on HPC architectures. It is an extremely useful optimisation tool! Our POP contact was very knowledgeable and enthusiastic. An excellent service!”

*Dr Joseph Parker, STFC UK*