Performance Optimisation and Productivity Of Parallel Codes

Samir.ben-Chaabane@teratec.eu

HORIZON-EUROHPC-JU-2023-COE

EuroHPC Joint Undertaking

1 January 2024– 31 December 2026

Grant Agreement No 101143931

POP COE

- 3rd Phase of the CoE (P1 2015-2018, P2 2019-2022)
- Promotes best practices in parallel programming
  - On **Performance Optimisation and Productivity** Enhancing parallel software can lead to significant benefits, including reduced costs, faster results, and innovative solutions.
  - Promoting **best practices in parallel programming**
  - Developing **the POP Methodology -** applying a structured approach for performance optimisation: Quantifying application behavior for targeted improvements.

- Free services for all EU academic and industrial codes and users
  - Precise understanding of application and system behaviour ➔ ***Performance Assessment***
  - Suggestion/support on how to refactor code in the most productive way : ***Second Level Service***
- **Horizontal**
  - Transversal across application areas, platforms, scales
- **For (EuroHPC) academic AND industrial codes and users !**

Partners

- **Who?**
  - BSC, ES (coordinator)
  - HLRS, DE
  - INESC-ID, PT
  - IT4I, CZ
  - JSC, DE
  - RWTH Aachen, IT Center, DE
  - TERATEC, FR
  - UVSQ, FR

A team with

- Excellence in performance tools and tuning
- Excellence in programming models and practices
- Research and development background AND proven commitment in application to real academic and industrial use cases

Motivation

Why?

- Complexity of machines and codes
  - ⇨ Frequent lack of quantified understanding of actual behaviour
  - ⇨ Not clear most productive direction of code refactoring

- Important to maximize efficiency (performance, power) of compute intensive applications and productivity of the development efforts

What?

- Parallel programs, mainly MPI/OpenMP
  - Although also CUDA, OpenCL, OpenACC, Python, …

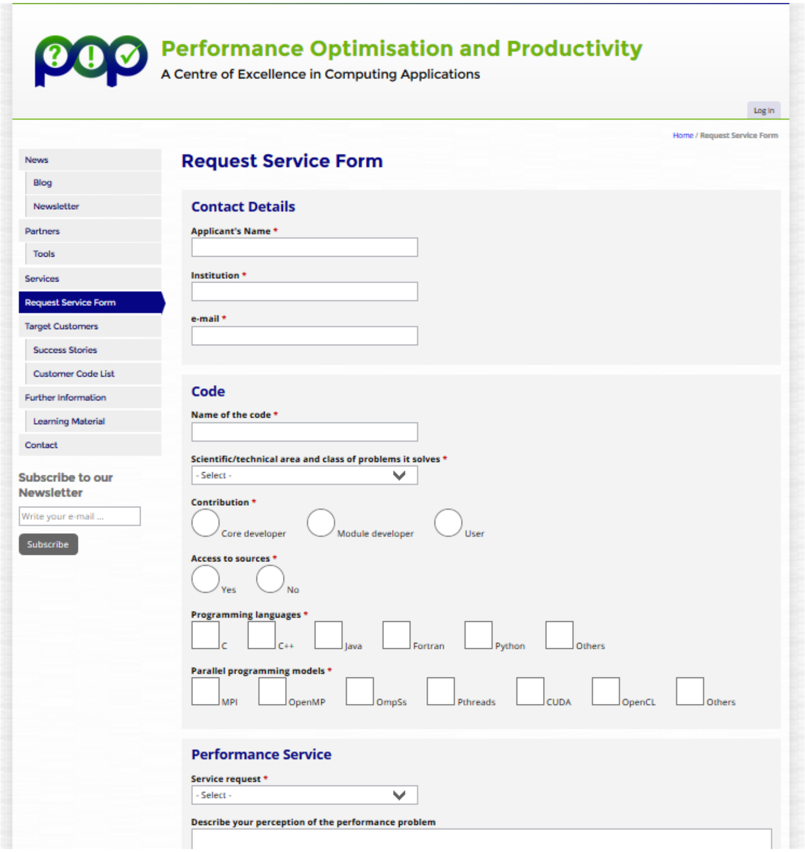
The Process …

When?

January 2024 – December 2026

How?

- Apply
  - Fill in small questionnaire describing application and needs
    https://pop-coe.eu/request-service-form
  - Questions? Ask pop@bsc.es
- Selection/assignment process
- Install tools @ your production machine (local, PRACE, …)
- Interactively: Gather data → Analysis → Report



FREE Services provided by the CoE

Parallel Application Performance Assessment	**Primary service** Initial analysis measuring a range of performance metrics to assess quality of performance and identify the issues affecting performance (at customer site) If needed, undertakes further performance evaluations to identify the root causes of the issues found and qualify and quantify approaches to address them (recommendations)
Second Level Services	Second level services may follow after conclusion of an initial performance assessment: • **Proof-of-concept**: explore the potential benefit of proposed optimisations by applying them to selected regions of the applications • **Correctness-check**: evaluate the correctness of hybrid MPI + OpenMP applications • **Energy-efficiency study**: investigate improvements of energy consumption or efficiency • **Advisory study**: ongoing consultancy for customers that choose to implement proposed optimisations on their own

*Note: Effort shared between our experts and customer!

Online Content

POP Website

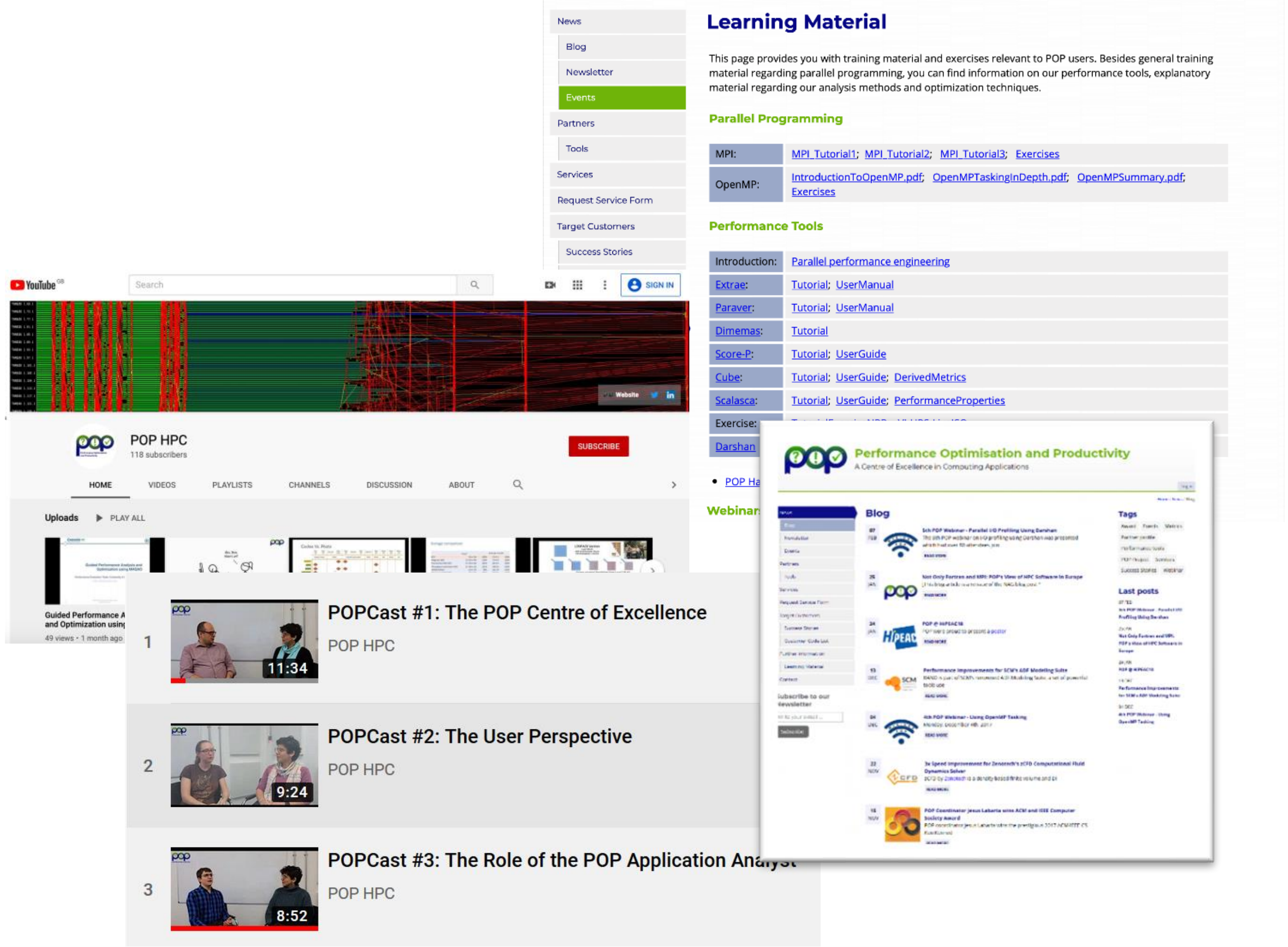
www.pop-coe.eu

- All the information you need to access POP services
  - **https://pop-coe.eu/services**
- Blogs
- More Learning Materials
- Newsletter
  - subscribe and see past issues

YouTube Channel

https://www.youtube.com/pophpc

- Past Webinars
- POPCasts



POP Online training course

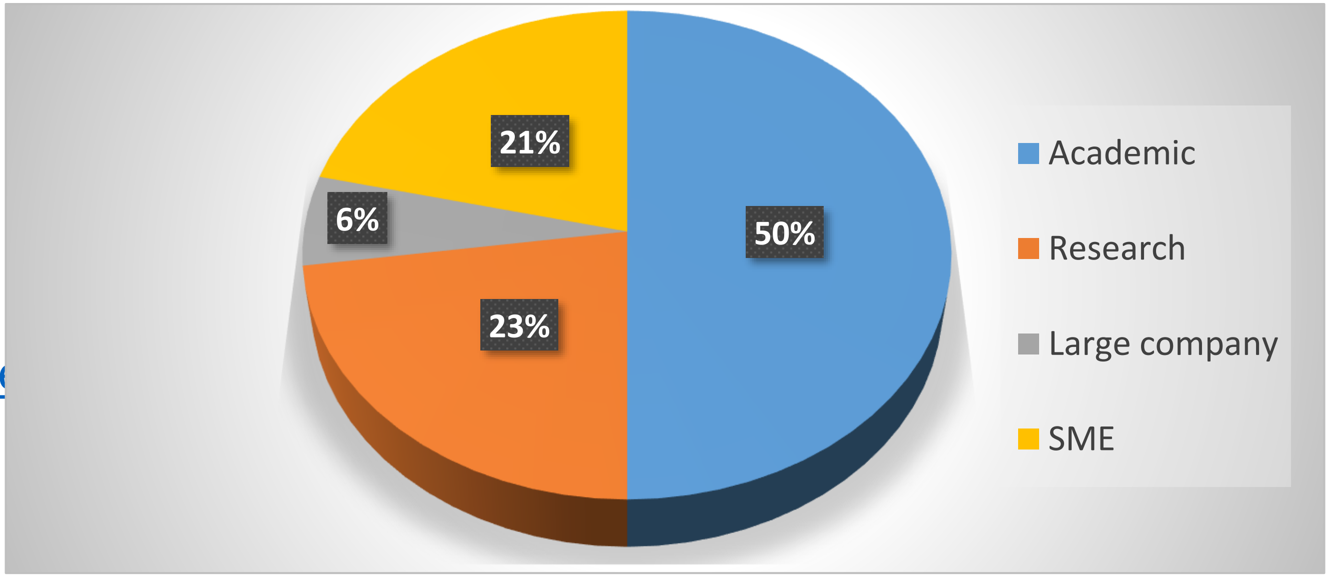
- A series of self-study modules
  - For those with limited experience in performance analysis of HPC applications
- Learning Objectives:
  - The challenges involved in HPC performance analysis
  - How the POP Metrics aid understanding of application performance
  - How to calculate the POP Metrics for your own HPC applications
  - What POP tools are available and how they can be installed
  - How to capture and analyse performance data with the POP tools

Target Customers

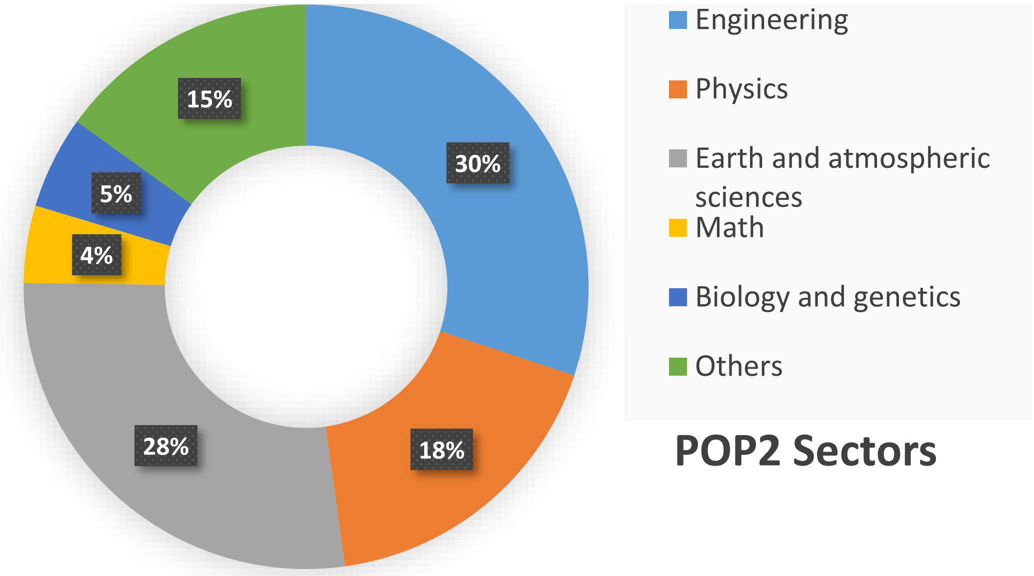
- **Code developers**
  - Assessment of detailed actual behaviour
  - Suggestion of most productive directions to refactor code
- **Users**
  - Assessment of achieved performance in specific production conditions
  - Possible improvements modifying environment setup
  - Evidence to interact with code provider
- **Infrastructure operators**
  - Assessment of achieved performance
    in production conditions
  - Possible improvements from modifying environment setup
  - Information for time computer time allocation processes
  - Training of support staff
- **Vendors**
  - Benchmarking
  - Customer support
  - System dimensioning/design

POP1-POP2 Outcomes

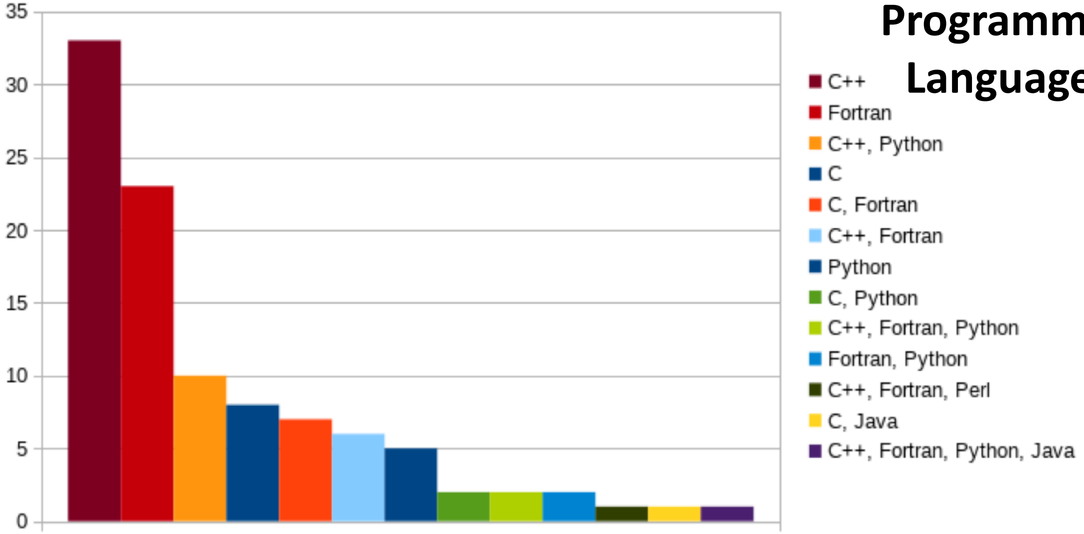
- **More than 250 services since Oct 2015** across all domains
  - E.g. engineering,
    earth & atmospheric sciences,
    physics, biology, genetics, …
  - **One service per week for 5 years now!**
- Outreach
  - More than **20 training workshops**
  - **19 webinars** so far
    See ⇨ https://pop-coe.eu/blog/tags/we



POP2 Services & HPC Codes

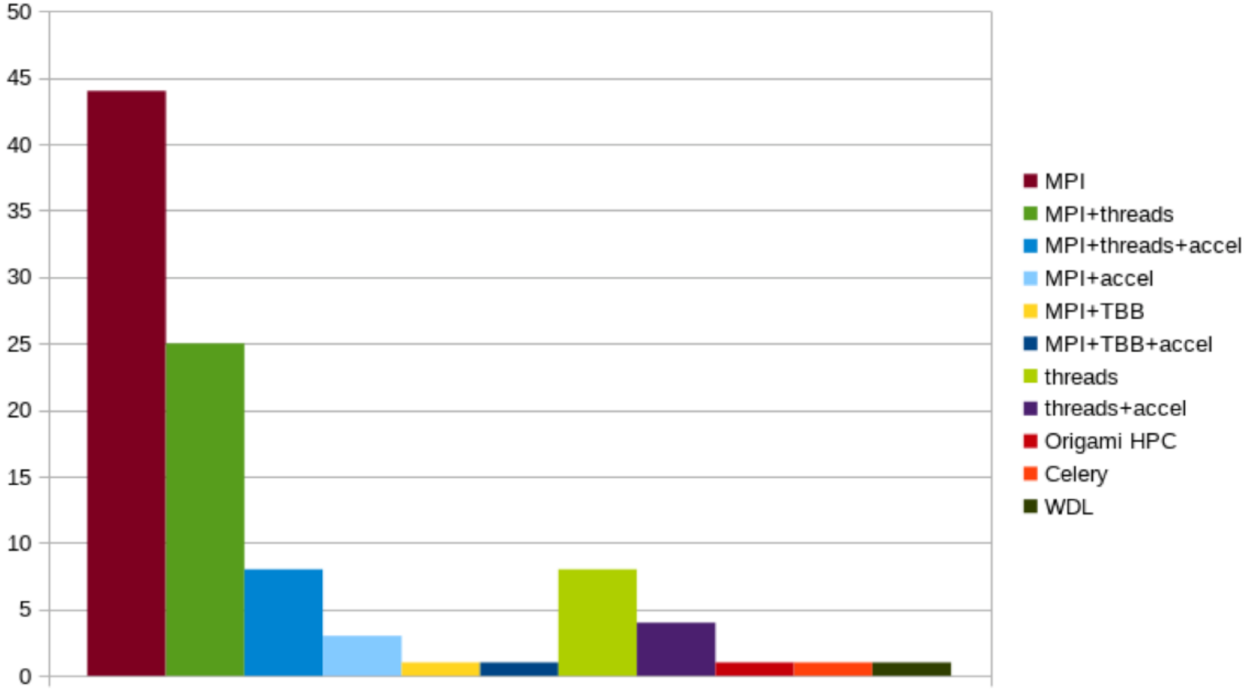


POP2 Sectors



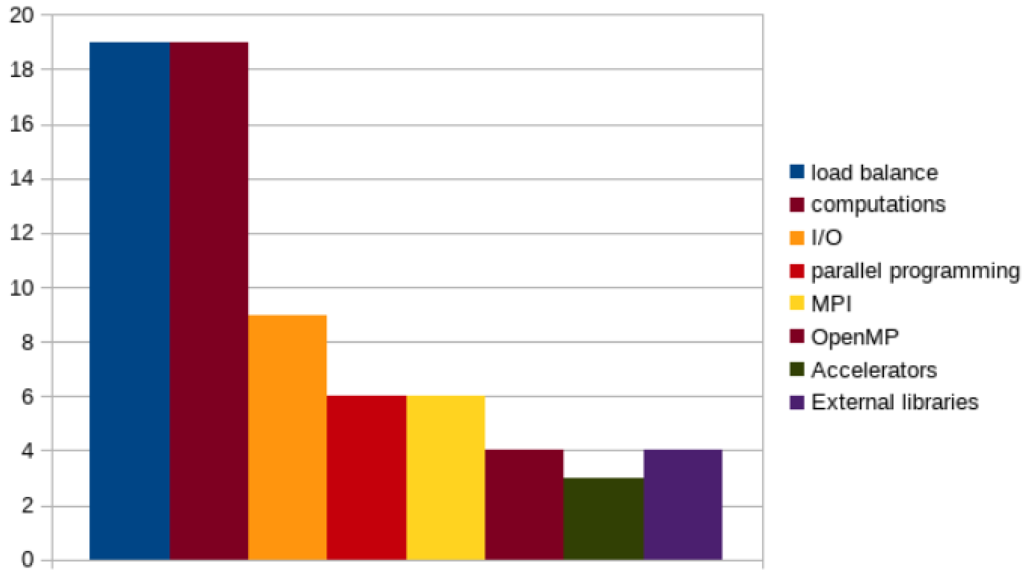
Programming Languages

Parallel Programming Models

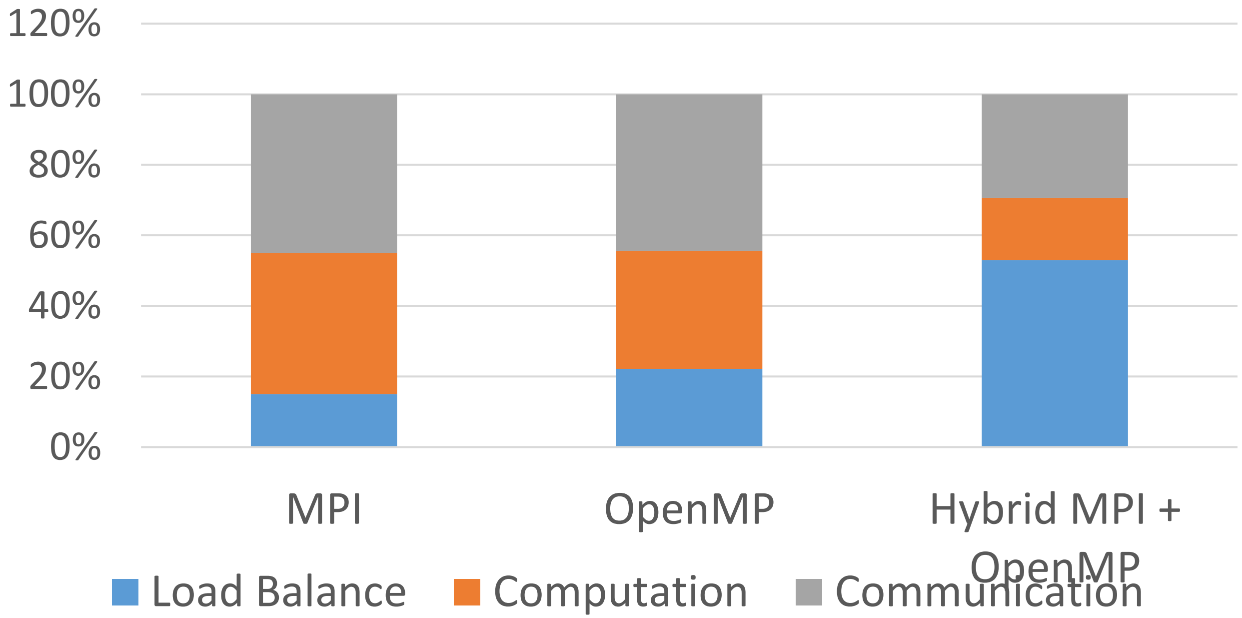


POP2 Services & HPC Codes (ctd)

Main sources of inefficiency



Inefficiency per type op parallélisation



ROI Examples

Application Savings after POP Proof-of-Concept

- POP PoC resulted in 72% faster-time-to-solution
- Production runs on ARCHER (UK national academic supercomputer)
- Improved code saves €15.58 per run
- Yearly savings of around €56,000 (from monthly usage data)

Application Savings after POP Performance Assessment

- Cost for customer implementing POP recommendations: €2,000
- Achieved improvement of 62%
- Resulted in yearly saving of €12,400 in compute costs ⇨ ROI of 620%

Example : A Computational Fluid Dynamics Code

- Code: **Fortran, OpenMP**
  - POP *Performance Assessment* followed by *Proof of Concept* service
- Platform:
  - MareNostrum-IV(@BSC)
    - Dual Intel Xeon Platinum 8160 Skylake 48-core nodes
- Tools used:
  - **Extrae & Paraver**
  - **Vtune**
  - **MAQAO**
- Scale:
  - **1-45 threads**

POP Metrics for the Performance assessment

# threads	1	2	10	18	30	45
Global Efficiency	1.00	0.86	0.65	0.41	0.31	0.15
↳ Parallel Efficiency	1.00	0.97	0.80	0.69	0.62	0.59
↳ OpenMP Region Efficiency	1.00	0.97	0.81	0.69	0.63	0.60
↳ Serial Region Efficiency	1.00	1.00	0.99	0.99	0.99	0.99
↳ Computational Scaling	1.00	0.89	0.81	0.60	0.49	0.26
↳ Instruction Scaling	1.00	1.00	1.00	1.00	0.99	0.97
↳ IPC Scaling	1.00	0.87	0.80	0.60	0.51	0.36
↳ Frequency Scaling	1.00	1.02	1.02	1.00	0.97	0.74

Poor scalability of the code is due to multiple factors:

- **OpenMP Region Efficiency** and **reducing IPC** are major limiting factors,
- Resulting in, respectively, poor Parallel Efficiency and poor Computational scaling

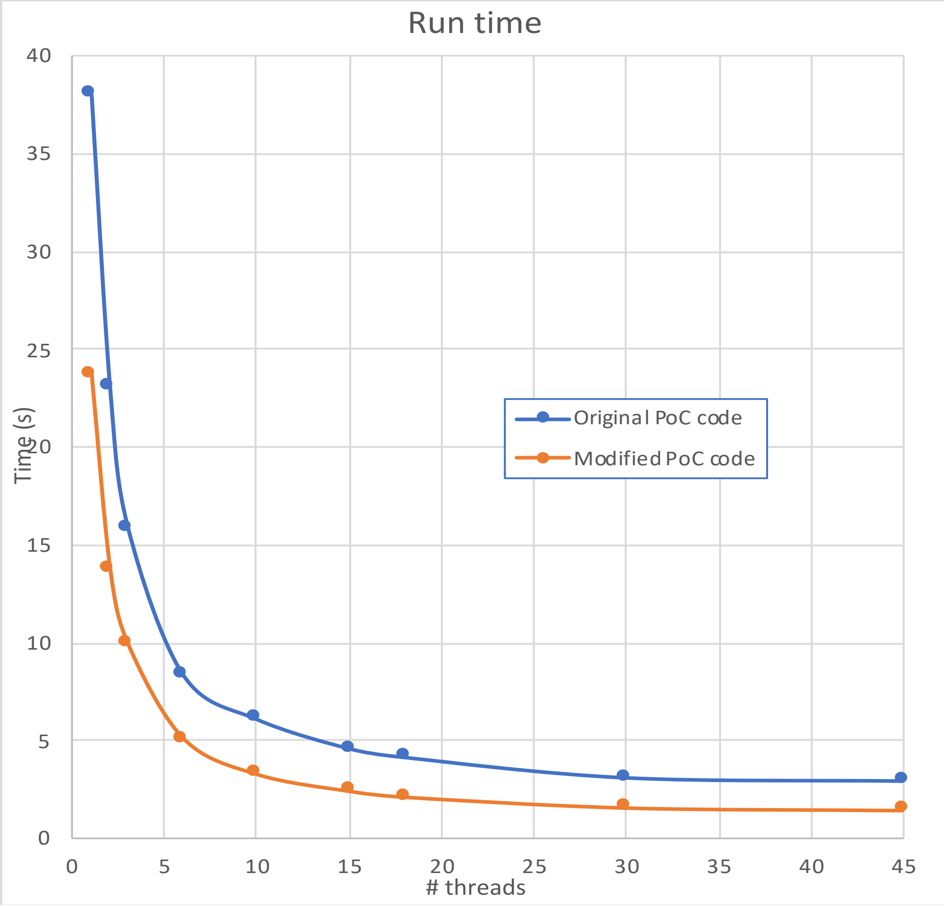
Example : Improving the Performance

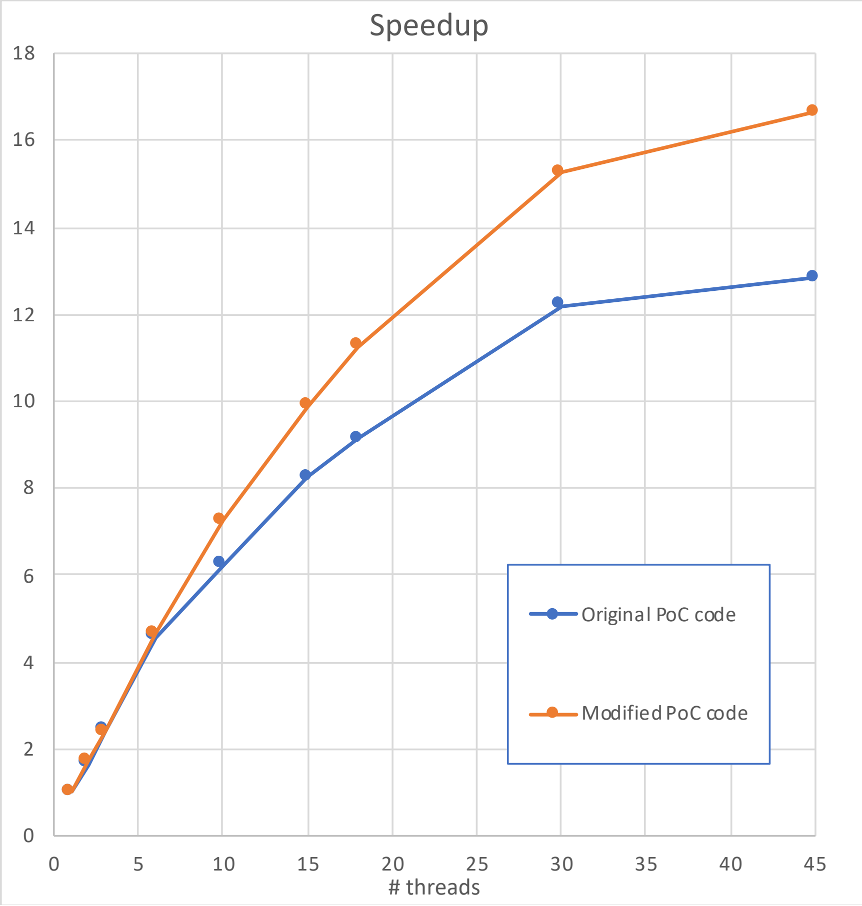
- Refactoring the code to address performance issues via POP Proof of Concept
  - Use of OpenMP COLLAPSE clause to improve load balance
  - Move some calculations outside the loops & remove unnecessary calculations
  - Use optimal loop ordering with nested loops

	Original code for *Proof of Concept*						Modified code					
# threads	1	2	10	18	30	45	1	2	10	18	30	45
Global Efficiency	1.00	0.86	0.65	0.41	0.31	0.15	1.00	0.86	0.72	0.62	0.51	0.37
↳ Parallel Efficiency	1.00	0.97	0.80	0.69	0.62	0.59	1.00	0.97	0.90	0.83	0.78	0.75
↳ OpenMP Region Efficiency	1.00	0.97	0.81	0.69	0.63	0.60	1.00	0.97	0.91	0.85	0.80	0.78
↳ Serial Region Efficiency	1.00	1.00	0.99	0.99	0.99	0.99	1.00	1.00	0.99	0.98	0.98	0.98
↳ Computational Scaling	1.00	0.89	0.81	0.60	0.49	0.26	1.00	0.88	0.81	0.75	0.65	0.49
↳ Instruction Scaling	1.00	1.00	1.00	1.00	0.99	0.97	1.00	1.00	1.00	0.99	0.99	0.98
↳ IPC Scaling	1.00	0.87	0.80	0.60	0.51	0.36	1.00	0.89	0.82	0.77	0.67	0.56
↳ Frequency Scaling	1.00	1.02	1.02	1.00	0.97	0.74	1.00	1.00	0.98	0.98	0.98	0.89

Example : Performance of modified code

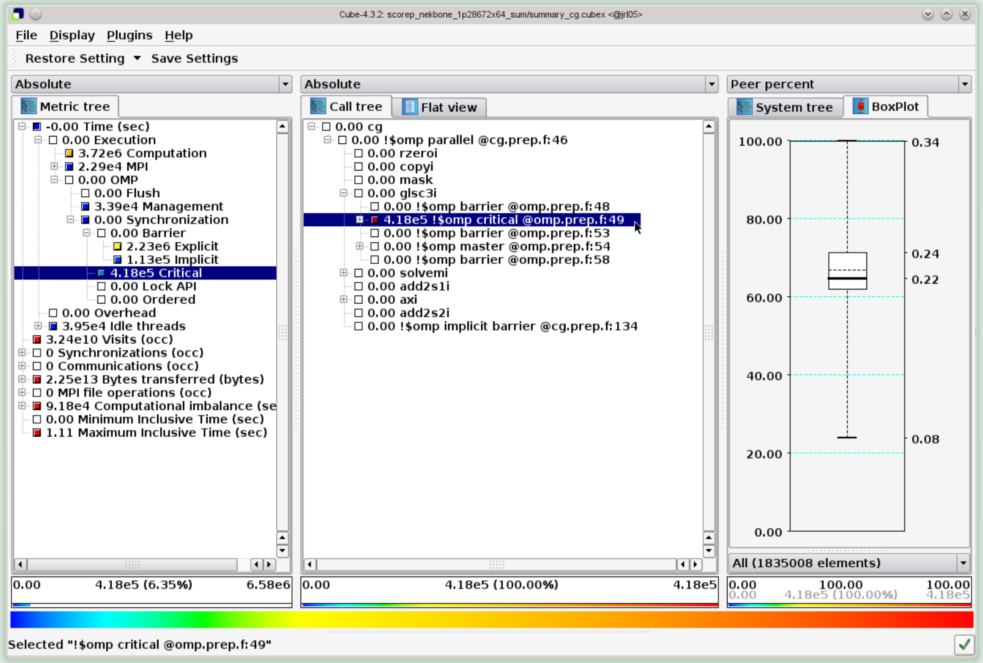
- The modified code
  - is 1.6x faster on 1 thread due to reduced instruction count
  - is 2.1x faster than original on 45 threads
  - shows better parallel scaling with a speedup of 16.7 on 45 threads relative to 1 thread

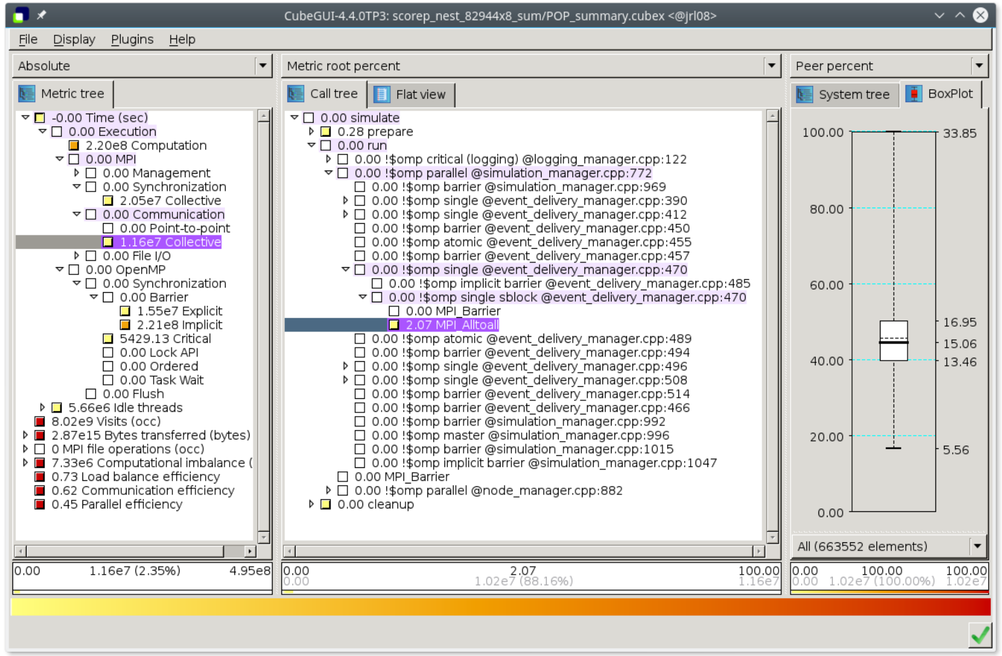




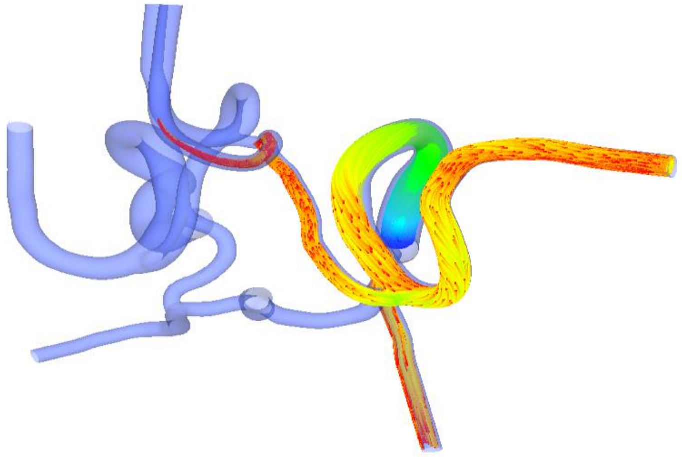
Exascale Readiness of Tools (Scalasca)

- Largest experiment by expert
  - Application: Nekbone
  - System: JuQueen BG/P
  - 28,672 x 64 = 1,835,008 threads
- Largest experiment by user @ RIKEN
  - Application: NEST
  - System: K computer
  - 82,944 x 8 = 663,552 threads



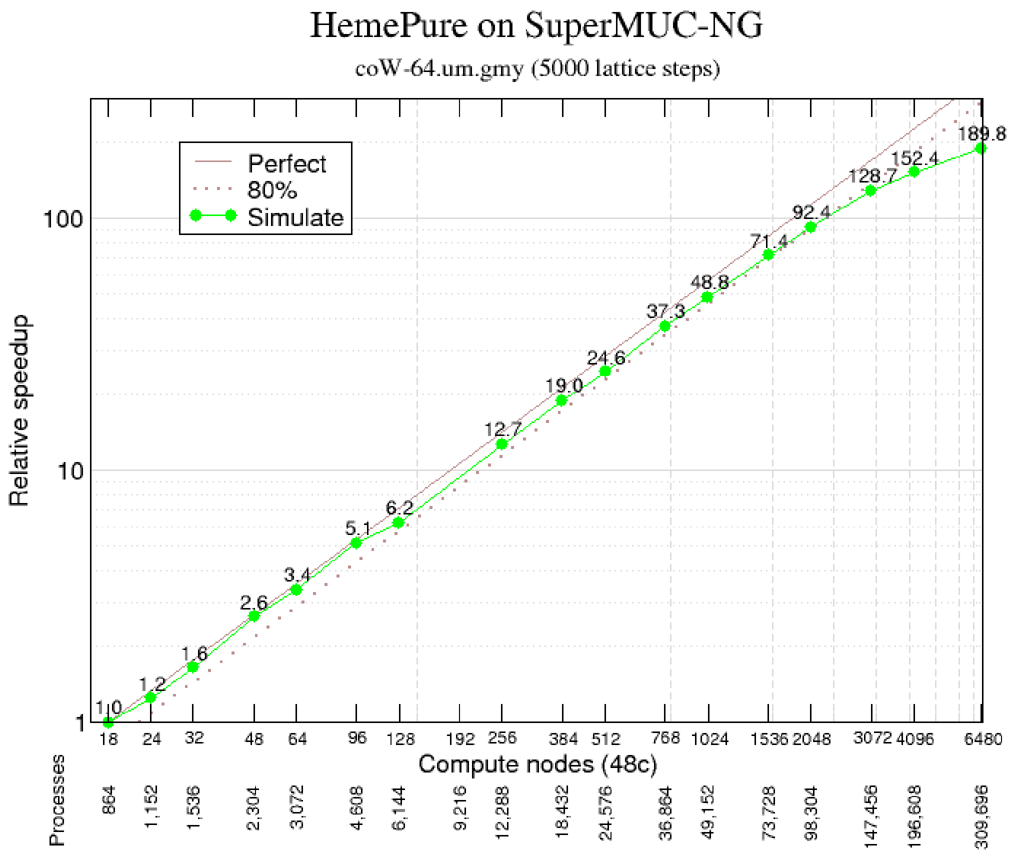


Exascale Readiness of Services



- **HemeLB**: open-source lattice-Boltzmann code for simulation of large-scale three-dimensional fluid flow in complex sparse geometries

- Developed within the EU H2020 HPC **CompBioMed** CoE
- POP assessment on SuperMUC (LRZ)
- Up to 309,696 MPI processes
- See ⇨ https://pop-coe.eu/blog/190x-strong-scaling-speed-up-of-hemelb-simulation-on-supermuc-ng



Some Success Stories

- More than 250 services since 2015 across all domains
  - e.g. engineering, earth & atmospheric sciences, physics, biology and genetics
- See https://pop-coe.eu/blog/tags/success-stories
  - Performance Improvements for SCM's ADF Modeling Suite
  - **3x Speed Improvement** for zCFD Computational Fluid Dynamics Solver
  - **25% Faster time-to-solution** for Urban Microclimate Simulations
  - **2x performance improvement** for SCM ADF code
  - Proof of Concept for BPMF leads to around **40% runtime reduction**
  - POP audit helps developers **double their code performance**
  - **10-fold scalability improvement** from POP services
  - POP performance study improves performance **up to a factor 6**
  - POP Proof-of-Concept study leads to **nearly 50% higher performance**
  - POP Proof-of-Concept study leads to **10X performance improvement** for customer

Summary

POP Performance Metrics

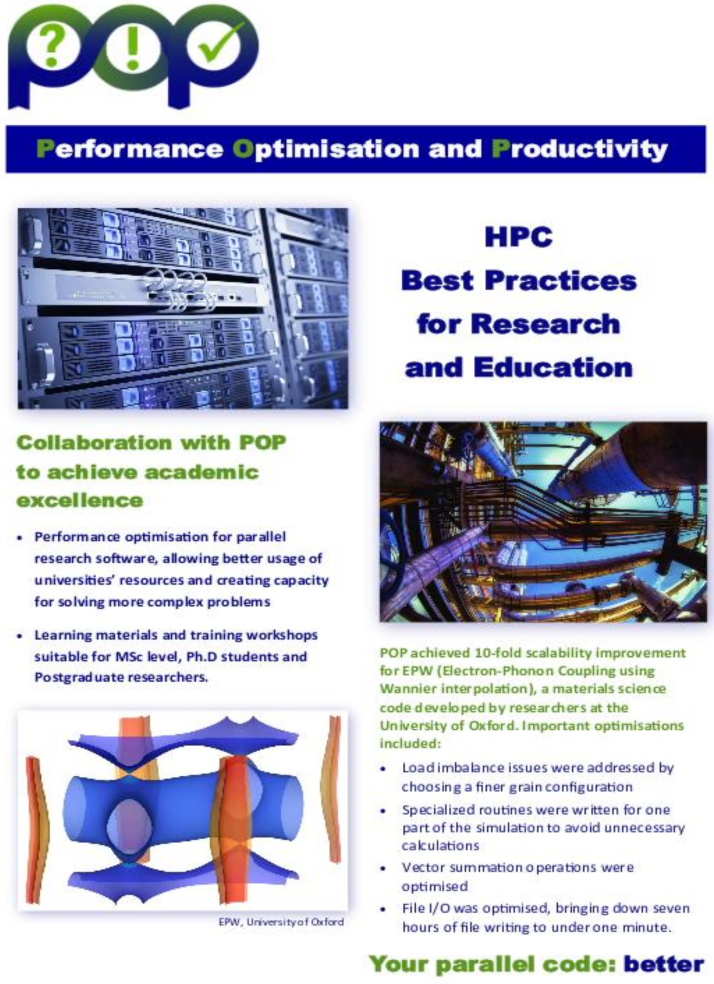
- Build a quantitative picture of application behavior
- Allow quick diagnosis of performance problems in parallel codes
- Identify strategic directions for code refactoring
- So far metrics for **MPI**, **OpenMP** and **Hybrid** (OpenMP + MPI) codes

POP works

- Across application domains, platforms, scales
- With (EU) academic and industrial customers including code developers, code users, HPC service providers and vendors
  - To apply for a POP service go to https://pop-coe.eu/services

POP CoE

- Promotes best practices in parallel programming
- Encourages a systematic approach to performance optimization
- Facilitates and invests in training HPC experts



EuroHPC
Joint Undertaking

What we expect from NCCs

- Promote the services provided by POP CoE
- Identify industrial users, mainly SMEs who can benefit from these services
- relay announcements for training courses and seminars

- For more information : https://pop-coe.eu/
- Samir.ben-Chaabane@teratec.eu

Performance Optimisation and Productivity 3

A Centre of Excellence in HPC

Contact:

https://www.pop-coe.eu

pop@bsc.es

@POP_HPC

youtube.com/POPHPC

This project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 101143931. The JU receives support from the European Union's Horizon Europe research and innovation programme and Spain, Germany, France, Portugal and the Czech Republic.

EuroHPC
Joint Undertaking