



We are the “Performance” in HPC

Bernd Mohr, POP CoE Dissemination Manager

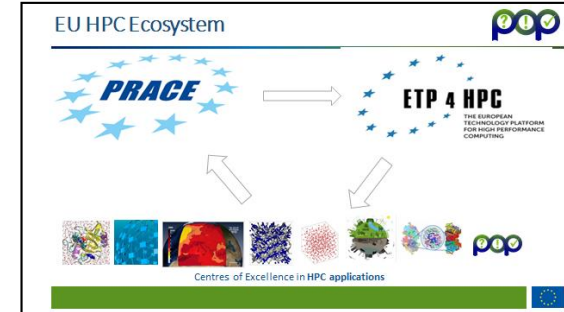
EU H2020 Centre of Excellence (CoE)



Grant Agreement No 824080

1 December 2018 – 30 November 2021

- A **Centre of Excellence**
 - On **Performance Optimisation and Productivity**
 - Promoting **best practices in parallel programming**
- Providing **FREE Services**
 - Precise understanding of application and system behaviour
 - Suggestion/support on how to refactor code in the most productive way
- **Horizontal**
 - Transversal across application areas, platforms, scales
- **For (EU) academic AND industrial codes and users !**



Partners



• Who?

- BSC, ES (coordinator)
- HLRS, DE
- IT4I, CZ
- JSC, DE
- NAG, UK
- 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

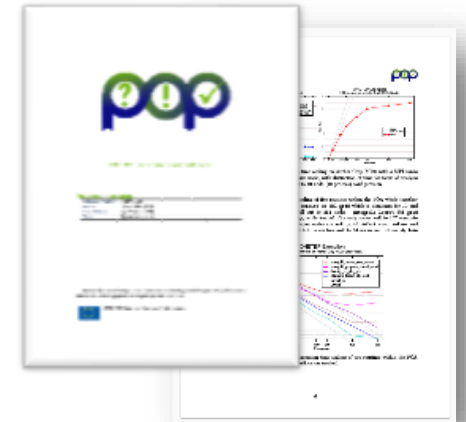


FREE Services provided by the CoE



- **Parallel Application Performance Assessment**

- Primary service
- Identifies performance issues of customer code (at customer site)
- If needed, identifies the root causes of the issues found and qualifies and quantifies approaches to address them (recommendations)
- **Combines former Performance Audit (?) and Plan (!)**
- Medium effort (1-3 months)



- **Proof-of-Concept (✓)**

- Follow-up service
- Experiments and mock-up tests for customer codes
- Kernel extraction, parallelisation, mini-apps experiments to show effect of proposed optimisations
- Larger effort (3-6 months)

```
<!DOCTYPE html>
<html id="home-layout">
  <head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8">
    <title>Source Code Pro</title>
    <!-- made with <3 and AFDKO -->
    <meta name="keywords" content="sans, monospace, open source, coding, font">
    <link rel="stylesheet" type="text/css" href="https://sourcecodepro.com/css/main.css">
  </head>
  <body>
    <div id="main">
```

Note: Effort shared between our experts and customer!





Overview of Codes Investigated

Status after 2½ Years (End of Phase1)



Performance Audits and Plans

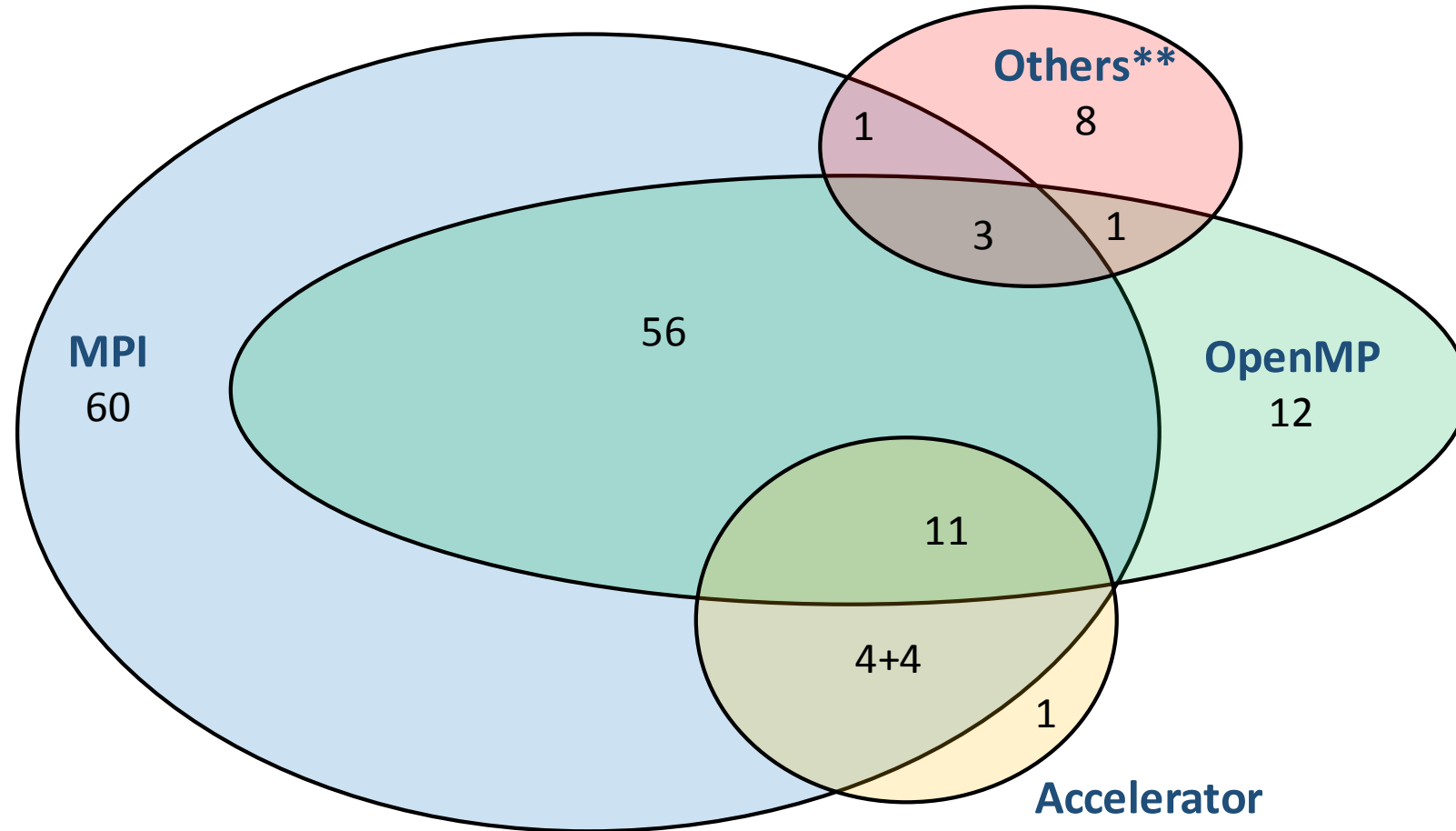
- 152 completed or reporting to customer

Proof-of- Concept

- 22 completed Proofs of Concept



Programming Models Used

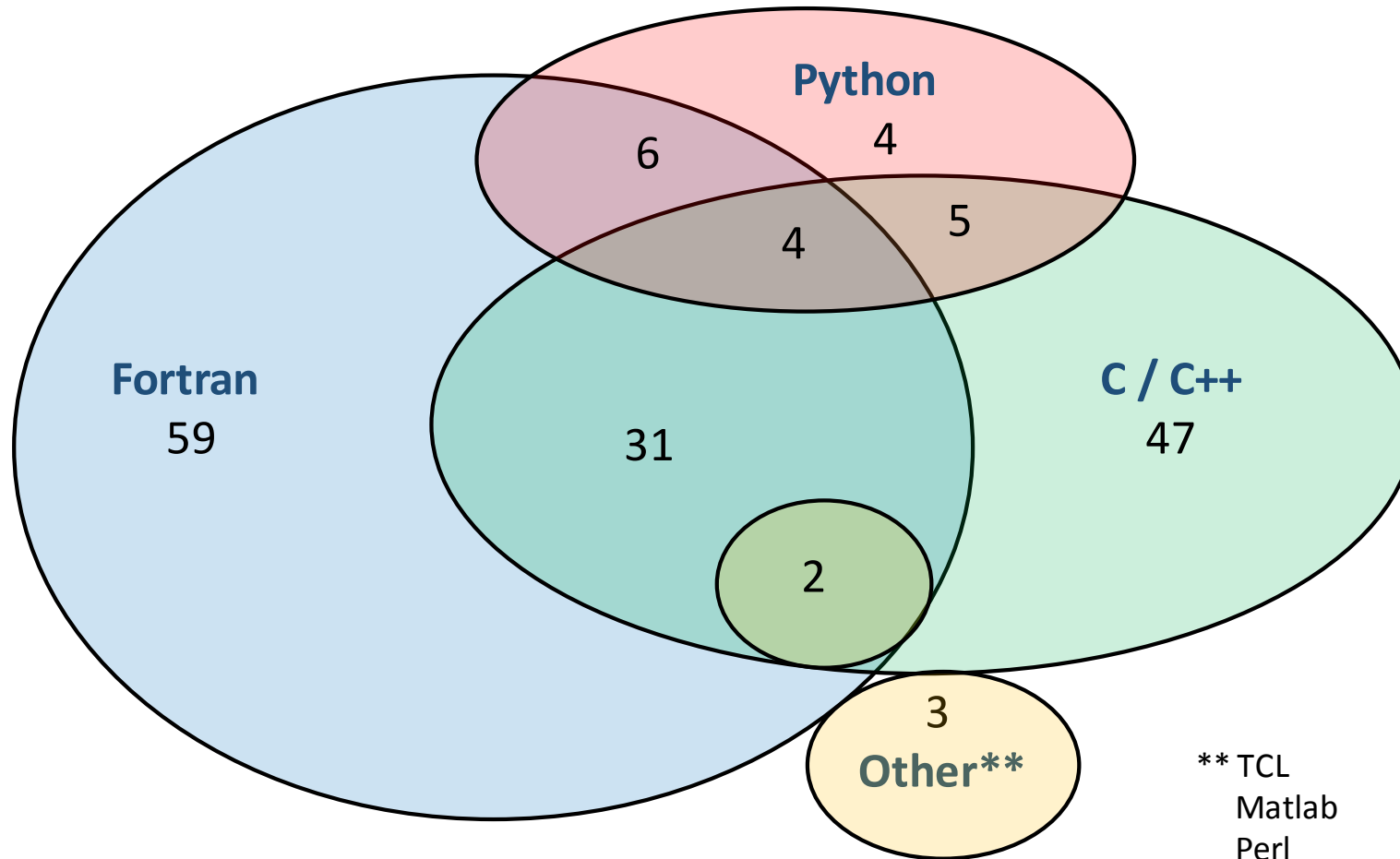


** MAGMA
Celery
TBB
GASPI
C++ threads
MATLAB PT
StarPU
GlobalArrays
Charm++
Fortran Coarray

* Based on data collected for 161 POP Performance Audits



Programming Languages Used



** TCL
Matlab
Perl
Octave
Java

* Based on data collected for 161 POP Performance Audits



Some PoC Success Stories



- See [⇒ https://pop-coe.eu/blog/tags/success-stories](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



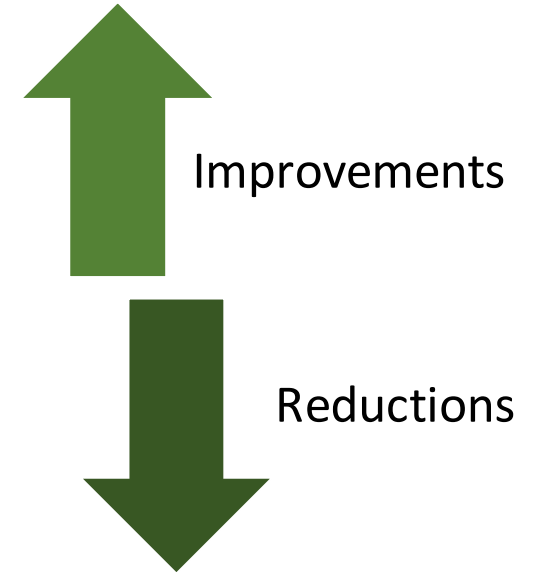
POP performance study improves performance **up to a factor 6**



POP Proof-of-Concept study leads to **nearly 50% higher performance**



POP Proof-of-Concept study leads to **10X performance improvement** for customer



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 Plan

- Cost for customer implementing POP recommendations: €2,000
- Achieved improvement of 62%
- €20,000 yearly operating cost
- Resulted in yearly saving of €12,400 in compute costs $\Rightarrow$ ROI of 620%

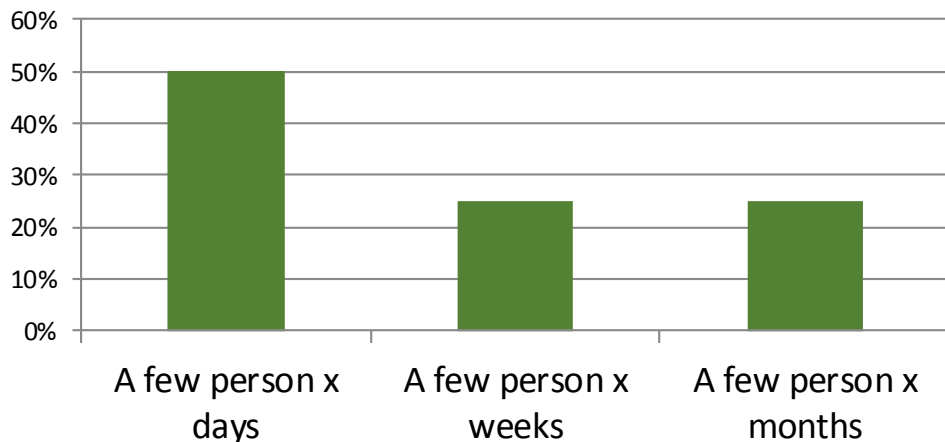
(Eight) Customers Success Feedback



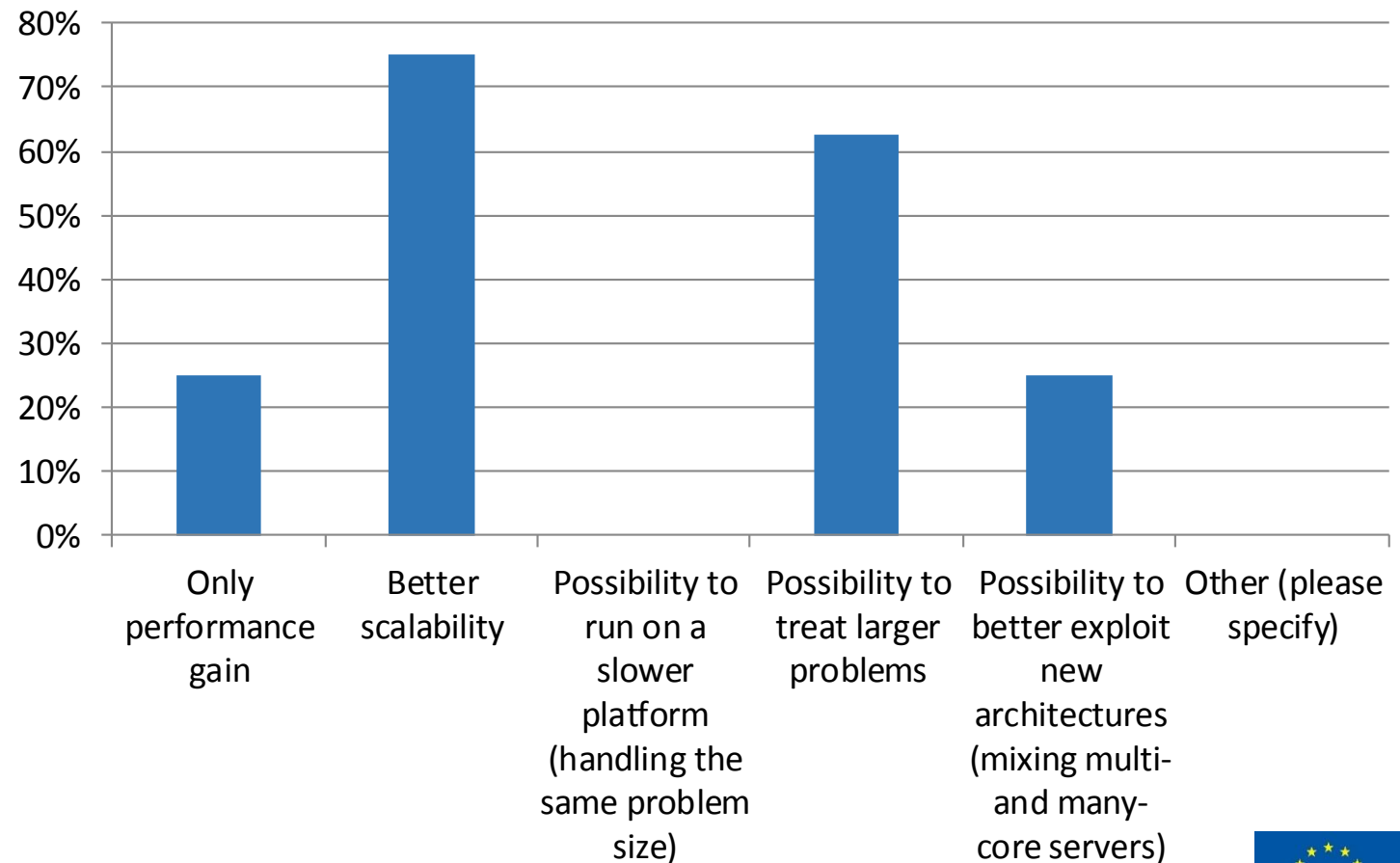
What is the observed performance gain after implementing recommendations?

25%
25%
20% overall, 50% for the given module
50-75% (case dependent)
12%
Up to 62 %, depending on the use case.
6 - 47 % depending on the test case.
15%

How much effort was necessary?



What are the main results?

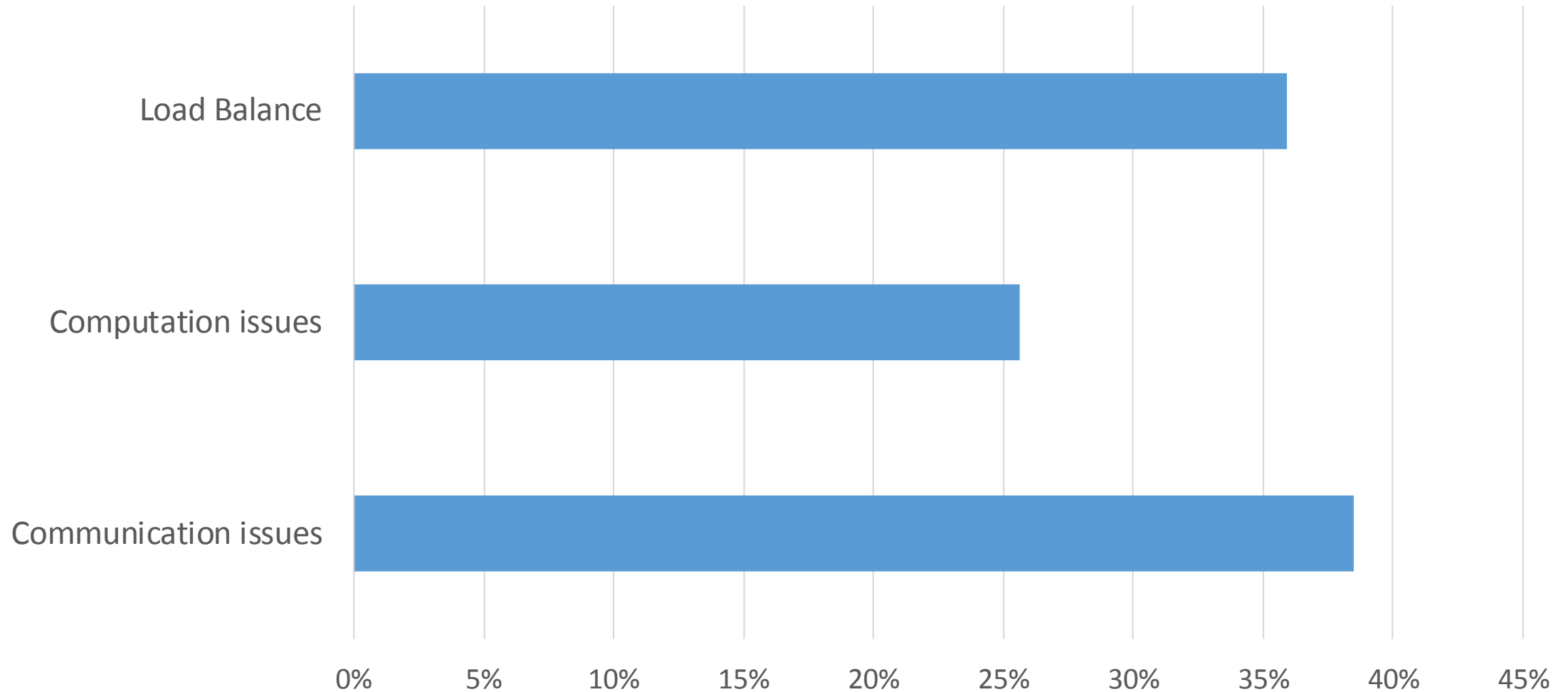




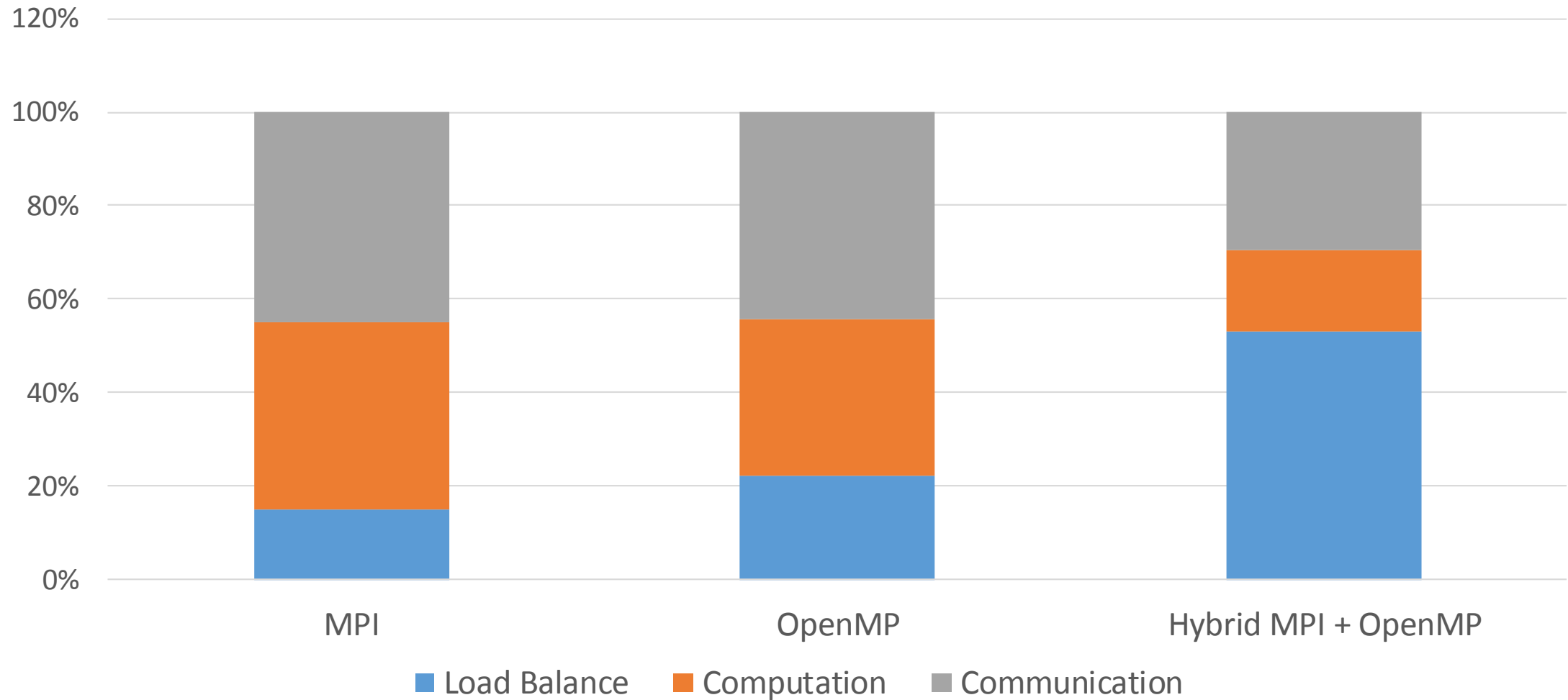
Analysis of Inefficiencies



Leading Cause of Inefficiency



Inefficiency by Parallelisation





Dissemination and Contact



Website – www.pop-coe.eu



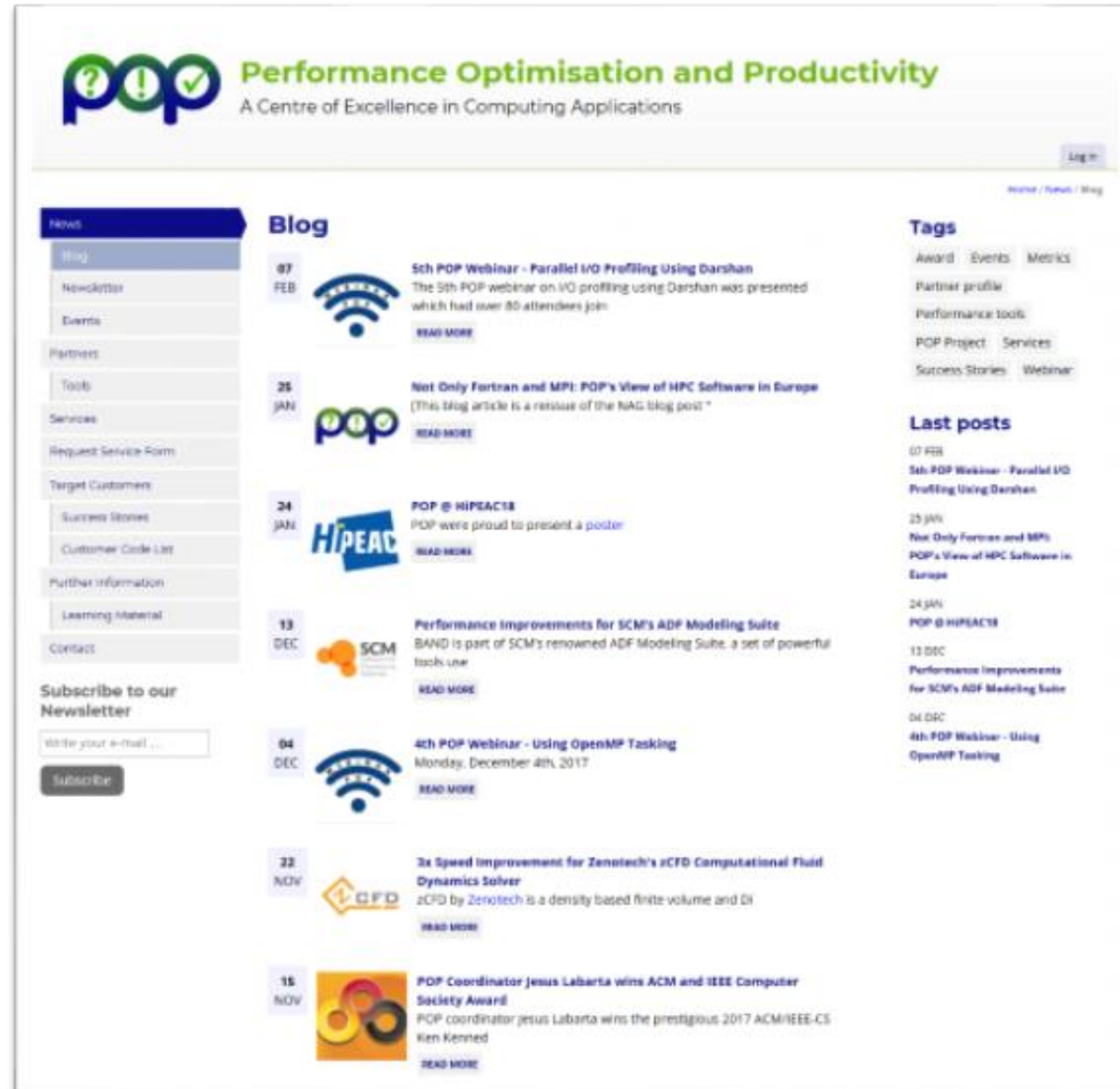
- POP User Portal
- Access to all public information and services

The screenshot shows the POP website homepage. At the top, the POP logo is on the left, followed by the text "Performance Optimisation and Productivity" and "A Centre of Excellence in HPC". A "Log in" button is in the top right. On the left side, there is a vertical menu with links: News, Blog, Newsletter, Events, Partners, Tools, Services, Request Service Form, Target Customers, Success Stories, Customer Code List, Further Information, Learning Material, Contact, and Privacy Policy. Below the menu is a "Subscribe to our Newsletter" section with an email input field and a "Subscribe" button. The main content area has a "Mission" section with text about performance optimisation and productivity services. Below that is a "Blog Highlights" section with three entries: "POP Project Restarted 1st December 2018", "Not Only Fortran and MPI: POP's View of HPC Software in Europe", and "A set of standard metrics for parallel performance analysis". On the right side, there is a "Latest News" section with a "POP RESTARTED Dec 1, 2018!" announcement, social media links for Twitter, YouTube, and LinkedIn, and a "Tweets" section showing recent tweets from @POP_HPC.

12-Dec-2018



- Typically 2 new articles per month
- Easy filtering via Tags, e.g.
 - Success Stories
 - Events
 - Webinars
 - ...



Follow us on Twitter @POP_HPC



Home Notifications Messages Search Twitter

POP_HPC
@POP_HPC
EU H2020 funded Centre of Excellence Performance Optimization and Productivity (POP) to boost performance and productivity in HPC applications.
pop-coe.eu
Joined October 2016
Born on October 1, 2000

TWEETS 26 FOLLOWING 145 FOLLOWERS 54 LIKES 5 MOMENTS 0

Tweets Tweets & replies Media

Pinned Tweet
POP_HPC @POP_HPC · Oct 19
Our aim is to help you optimise your parallel code. Do bigger, better, faster science with POP. pop-coe.eu

Who to follow · Refresh · View all

- EUROfusion** @EurofusionCI · Follow
- Horizon 2020** @Horizon2020 · Follow
- martorellBSC** @martorellBSC · Follow



Quarterly Email Newsletter



- Subscribe on POP website
- Newsletter archive: <https://pop-coe.eu/news/newsletter>



- See ➡ <https://pop-coe.eu/blog/tags/webinar>
- Or see our  [YouTube Channel](#)
- Already available:
 - How to Improve the Performance of Parallel Codes
 - Getting Performance from OpenMP Programs on NUMA Architectures
 - Understand the Performance of your Application with just Three Numbers
 - Using OpenMP Tasking
 - Parallel I/O Profiling Using Darshan
 - The impact of sequential performance on parallel codes
 - Large scale Application Execution Performance Assessment
 - POP Case Study: 3x Speed Improvement for Zenotech's zCFD Solver
 - Exascale Matrix Factorization: Using HPC and ML for Drug Discovery





Summary & Conclusion



- **PRACE Users**

- Preparing a large-scale project proposal for future PRACE call?
 - ⇒ Get a POP Assessment so you have an independent confirmation your code is efficient and scalable!
- Come to our performance analysis and tuning events
 - Some part of PRACE Training Centres

- **PRACE HLST**

- Talk to us about our performance analysis and tuning
 - Training
 - Support
 - Methodology

Summary & Conclusion



- **POP CoE Phase 1 finished in March 2018** after 2½ years
 - **Successfully demonstrated expertise and impact**
 - 152 Audits + Perf Plans / 22 Proof-of-Concept / 21 requests cancelled
 - **Intensive dissemination** via website, blog articles, tweets, newsletter, ...
- **POP CoE Phase 2 restarted in December 2018** for 3 more years
 - New Service Structure (Performance Assessment combines Audit and Plan)
 - New Project Partners (IT4I, UVSQ)
 - New Co-design Data Repository





Performance Optimisation and Productivity

A Centre of Excellence in HPC

Contact:

<https://www.pop-coe.eu>

<mailto:pop@bsc.es>

 @POP_HPC





Backup



The Process ...



When?

December 2018 – November 2021

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

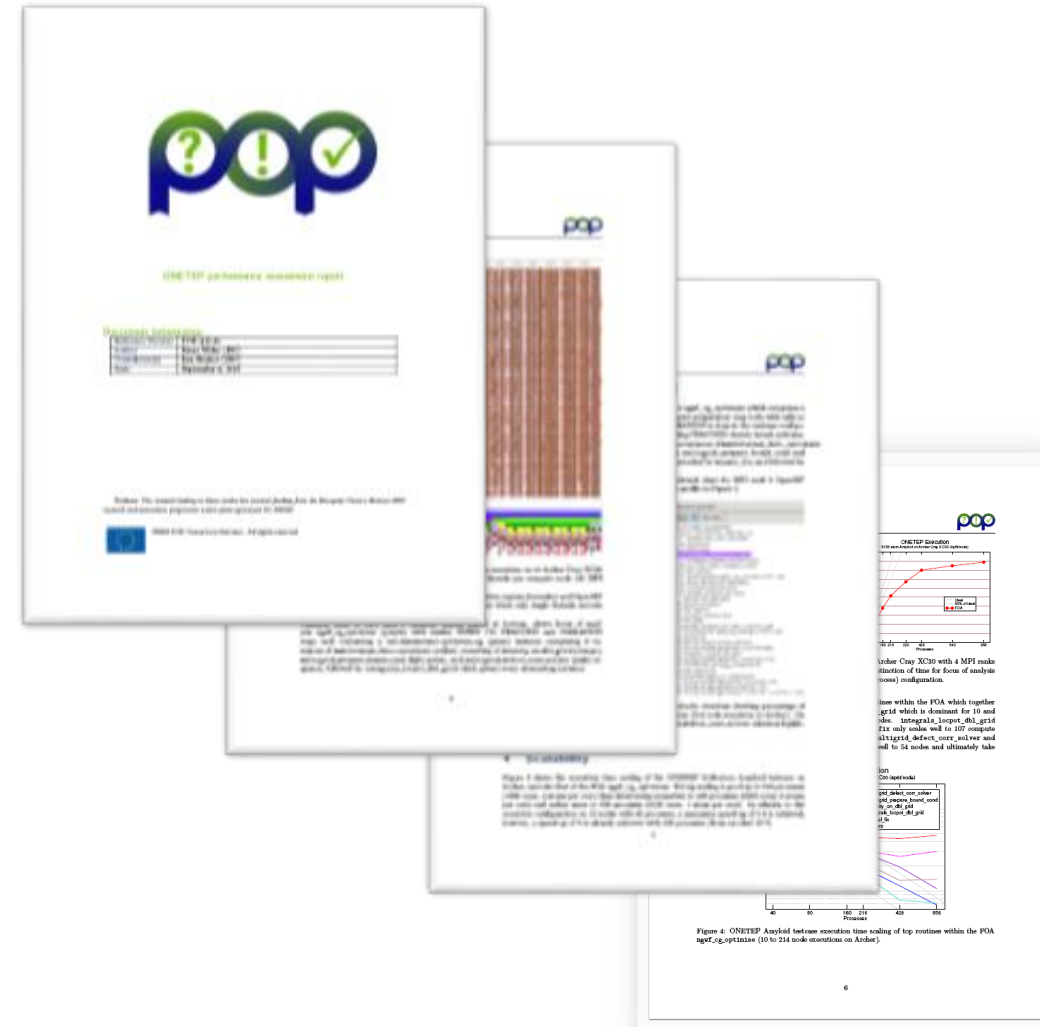
The screenshot shows the 'Request Service Form' on the POP website. The header includes the POP logo and the text 'Performance Optimisation and Productivity - A Centre of Excellence in Computing Applications'. A sidebar on the left contains links to News, Blog, Newsletter, Partners, Tools, Services, Request Service Form (highlighted), Target Customers, Success Stories, Customer Code List, Further Information, Learning Material, and Contact. Below the sidebar is a 'Subscribe to our Newsletter' section with an email input field and a 'Subscribe' button. The main form area is titled 'Request Service Form' and contains several sections: 'Contact Details' with fields for 'Applicant's Name', 'Institution', and 'e-mail'; 'Code' section with a 'Name of the code' field, a dropdown for 'Scientific/technical area and class of problems it solves', and radio buttons for 'Contribution' (Core developer, Module developer, User); 'Access to sources' with radio buttons for 'Yes' and 'No'; 'Programming languages' with checkboxes for C, C++, Java, Fortran, Python, and Others; 'Parallel programming models' with checkboxes for MPI, OpenMP, OpenMPs, Pthreads, CUDA, OpenCL, and Others; and 'Performance Service' with a dropdown for 'Service request' and a text area for 'Describe your perception of the performance problem'.



Outline of a Typical Audit Report



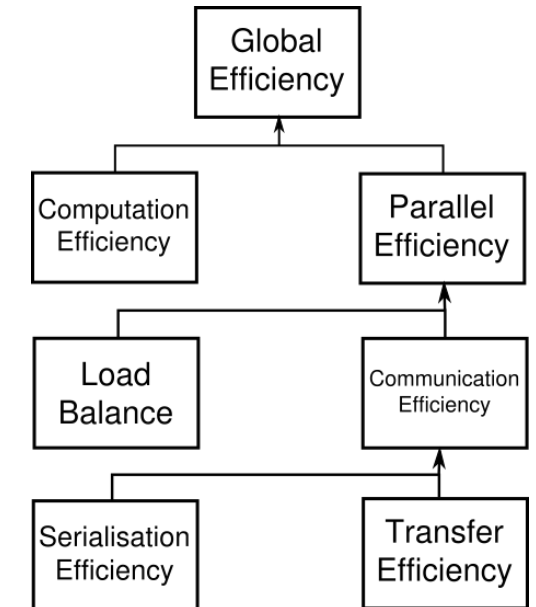
- Application Structure
- (If appropriate) Region of Interest
- Scalability Information
- Application Efficiency
 - E.g. time spent outside MPI
- Load Balance
 - Whether due to internal or external factors
- Serial Performance
 - Identification of poor code quality
- Communications
 - E.g. sensitivity to network performance
- Summary and Recommendations



Efficiencies



- The following metrics are used in a POP Performance Audit:
- Global Efficiency (GE): $GE = PE * CompE$
 - Parallel Efficiency (PE): $PE = LB * CommE$
 - **Load Balance** Efficiency (LB): $LB = avg(CT)/max(CT)$
 - **Communication** Efficiency (CommE): $CommE = SerE * TE$
 - Serialization Efficiency (SerE):
 $SerE = max(CT / TT \text{ on ideal network})$
 - Transfer Efficiency (TE): $TE = TT \text{ on ideal network} / TT$
 - (Serial) **Computation** Efficiency (CompE)
 - Computed out of IPC Scaling and Instruction Scaling
 - For strong scaling: ideal scaling -> efficiency of 1.0
- Details see <https://sharepoint.ecampus.rwth-aachen.de/units/rz/HPC/public/Shared%20Documents/Metrics.pdf>



CT = Computational time
TT = Total time

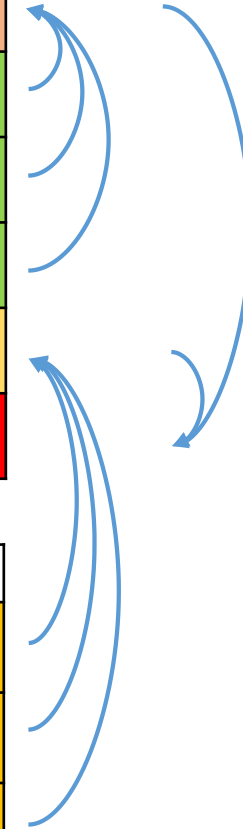


Efficiencies



	2	4	8	16
Parallel Efficiency	0.98	0.94	0.90	0.85
Load Balance	0.99	0.97	0.91	0.92
Serialization efficiency	0.99	0.98	0.99	0.94
Transfer Efficiency	0.99	0.99	0.99	0.98
Computation Efficiency	1.00	0.96	0.87	0.70
Global efficiency	0.98	0.90	0.78	0.59

	2	4	8	16
IPC Scaling Efficiency	1.00	0.99	0.96	0.84
Instruction Scaling Efficiency	1.00	0.97	0.94	0.91
Core frequency efficiency	1.00	0.99	0.96	0.91



- **Install and use already available monitoring and analysis technology**
 - Analysis and predictive capabilities
 - Delivering insight
 - With extreme detail
 - Up to extreme scale
- **Open-source toolsets**
 - Extrae + Paraver
 - Score-P + Cube + Scalasca/TAU/Vampir
 - Dimemas, Extra-P
 - MAQAO
- **Commercial toolsets**
(if available at customer site)
 - Intel tools
 - Cray tools
 - ARM 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

