Conference or Workshop Item:
Smith, Matthew A., Wright, Steven A. orcid.org/0000-0001-7133-8533, Lantra, Zaman et al. (1 more author) (2024) The P3 Explorer: Exploring the Performance, Portability, and Productivity Wilderness. In: The International Conference for High Performance Computing, Networking, Storage, and Analysis, 17-22 Nov 2024.

The P3 Explorer: Exploring the Performance, Portability, and Productivity Wilderness

Matthew A. Smith[1], Steven A. Wright[1], Zaman Lantra[2], Gihan Mudalige[2]
[1]University of York, UK, [2]University of Warwick, UK

Introduction

The P3 Explorer is an open data repository and visualisation tool, designed to showcase Performance, Portability, and Productivity (P3) data with plots and insights automatically generated by Intel's P3 Analysis Library [1].

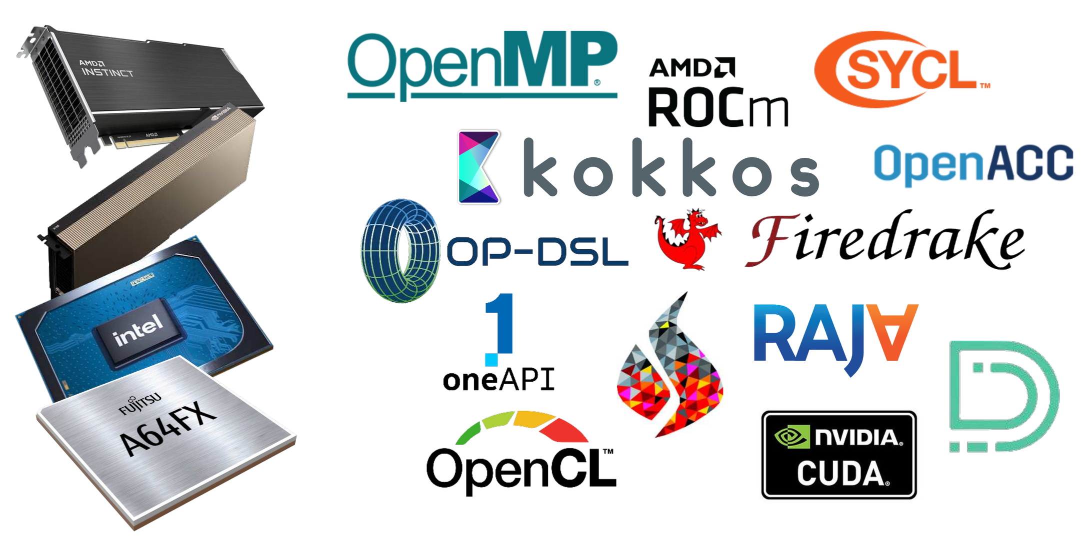
This provides the following benefits to the wider HPC community:

- A repository of user-submitted experimental data focused on the performance, portability, and productivity of HPC applications.
- A dashboard for exploring user-submitted data from P3 studies, using automatically-generated plots for visualising *performance portability,* and *productivity*.

The P3 Explorer provides an interface for application developers to assess a variety of approaches to developing new parallel software with the goal of maximising the *three Ps*.

Context

HPC hardware is diversifying, and there is a proliferation of parallel programming models that target heterogenous systems.



Developers are typically seeking to maximise ***Performance***, ***Portability***, and ***Productivity*** on modern heterogeneous HPC platforms [2].

Evaluating the Three Ps

The P3 Explorer uses the P3 Analysis Library to evaluate *performance portability* using the Pennycook metric [3].

$$\text{Ꝑ}(a, p, H) = \begin{cases} \dfrac{|H|}{\sum_{i \in H} \dfrac{1}{e_i(a, p)}} & \text{if } i \text{ is supported } \forall i \in H \\ 0 & \text{otherwise} \end{cases}$$

The library uses *code divergence* as a measure of developer productivity [4].

$$CD(a, p, H) = \binom{|H|}{2}^{-1} \sum_{\{i,j\} \in H \times H} d_{i,j}(a, p)$$

$$d_{i,j}(a, p) = 1 - \frac{|c_i(a, p) \cap c_j(a, p)|}{|c_i(a, p) \cup c_j(a, p)|}$$

To provide a more holistic view of performance portability and developer productivity, our dashboard generates *performance portability cascade plots* and *performance portability code convergence* charts (Ꝑ-CC) [5]. These allow developers to better evaluate the P3 properties of an application visually.

The P3 Explorer

The P3 Explorer primarily serves as a data dashboard, allowing application developers to explore P3 data and visualisations from previously published performance studies.

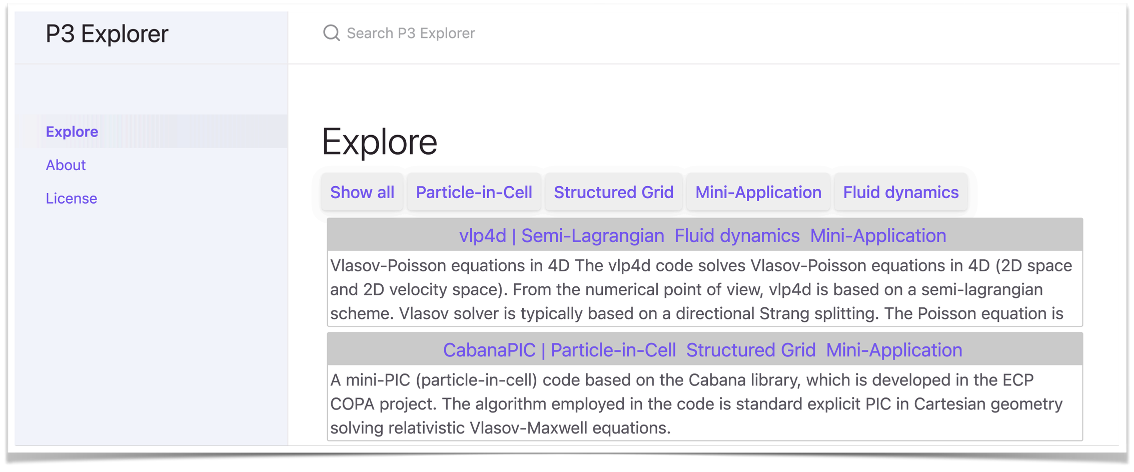


Figure 1: An example of the database exploration page

Performance studies are archived on the dashboard organised by application, and categorised by scientific domain(s) (**Figure 1**). Each study contains performance data, and a submission information file, providing metadata (e.g., paper DOI, links to source code). Each application may contain multiple studies, and no attempt is made to combine datasets.

The dashboard is built using Jekyll and deployed through GitHub Pages.

An Example

Each scientific study data page contains:

- Metadata related to the study, including a description of how the data was gathered.
- A bar chart of submitted performance data, plotted using Matplotlib.
- Appropriate P3 plots, generated using the P3 Analysis Library [1].

An example for CabanaPIC is shown below (**Figure 2**). From this study, the P3 properties of the OP-PIC DSL can be compared to Kokkos, showing no code divergence in either case and better performance portability for OP-PIC.

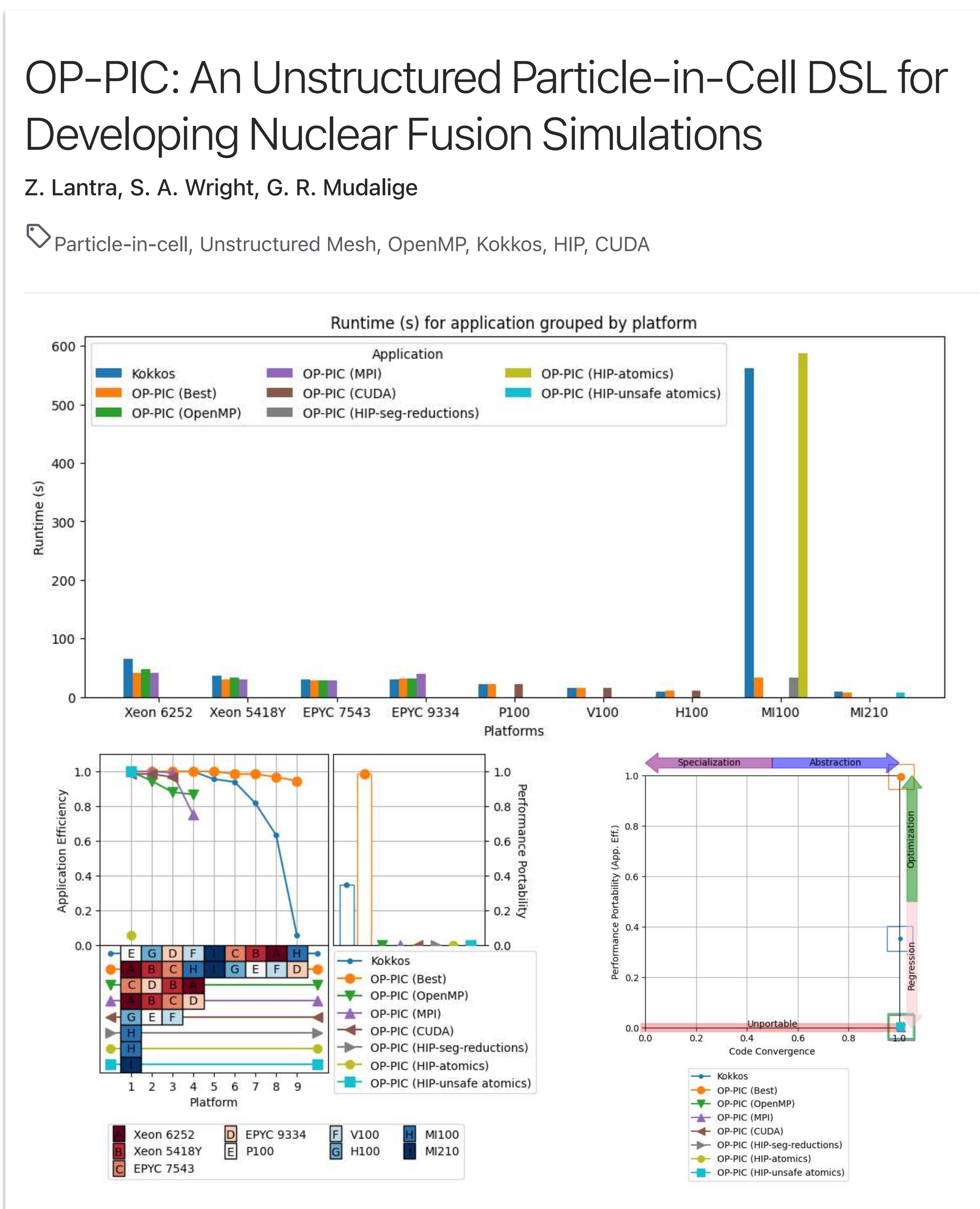


Figure 2: An example experimental data page for Lantra et al. [6]

Contributing

Contributing to the project is simple, and all of the data processing is done automatically on merging a *pull request*.

To contribute data, a pull request should be created that adds data to the *submissions* directory (**Figure 3**).

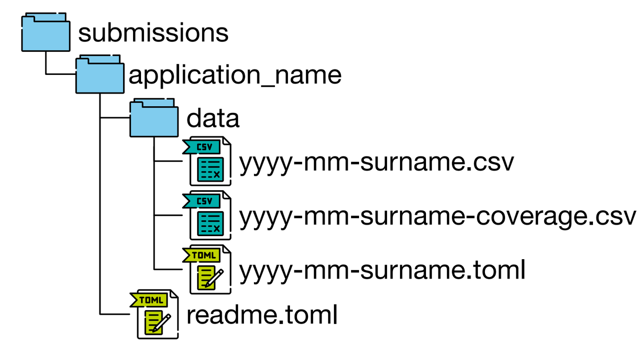


Figure 3: An illustration of the submissions directory structure

Data submissions should include performance data in CSV format, and a TOML file with information about the submission (**Figure 4**). An additional readme file is required for each *unique* application with high-level detail about the application.

```
title       = "OP-PIC: An Unstructured Particle-in-Cell DSL ..."
authors     = ["Z. Lantra" , "S. A. Wright", "G. R. Mudalige"]
sources     = [
                ["OP-PIC Repository", ...],
                ["OP-PIC Documentation", ...]
              ]
doi         = "10.1145/3673038.3673130"
fom         = "Runtime (s)"
tags        = ["Particle-in-cell", ...]
description = """In this work we introduce OP-PIC ..."""
```

Figure 4: An example TOML file for a data submission

Optional code coverage data can be provided in CSV format to generate a Ꝑ-CC chart.

Datasets can be updated at any time. The dashboard updates through CI/CD actions on a commit or merge.

Highlights

Performance, portability, and productivity are vital when designing and developing new HPC applications for heterogeneous systems.

- Our repository provides an open archive of scientific data from past studies focused on performance, portability, and productivity.
- Our visualisation dashboard can be used by developers to help inform application design decisions to better achieve performance, portability, and productivity.
- Our dashboard encourages full disclosure of configuration data, performance data, and P3 analysis to improve the reproducibility of performance studies.

References

[1] S. J. Pennycook, et al. (2023). *Performance, Portability and Productivity Analysis Library*. https://github.com/intel/p3-analysis-library

[2] S. A. Wright, et al. (2024) *Developing performance portable plasma edge simulations: A survey*. Computer Physics Communications, 298

[3] S. J. Pennycook, et al. (2019) *Implications of a metric for performance portability*. Future Generation Computer Systems, 92

[4] S. L. Harrell, et al. (2018) *Effective performance portability*. Proceedings of the IEEE/ACM International Workshop on Performance, Portability and Productivity

[5] S. J. Pennycook, et al. (2021) *Navigating Performance, Portability, and Productivity*. Computing in Science & Engineering, 23(5)

[6] Z. Lantra, et al. (2024) *OP-PIC: An Unstructured Particle-in-Cell DSL for Developing Nuclear Fusion Simulations*. Proceedings of the International Conference on Parallel Processing

Department of Computer Science
University of York
york.ac.uk