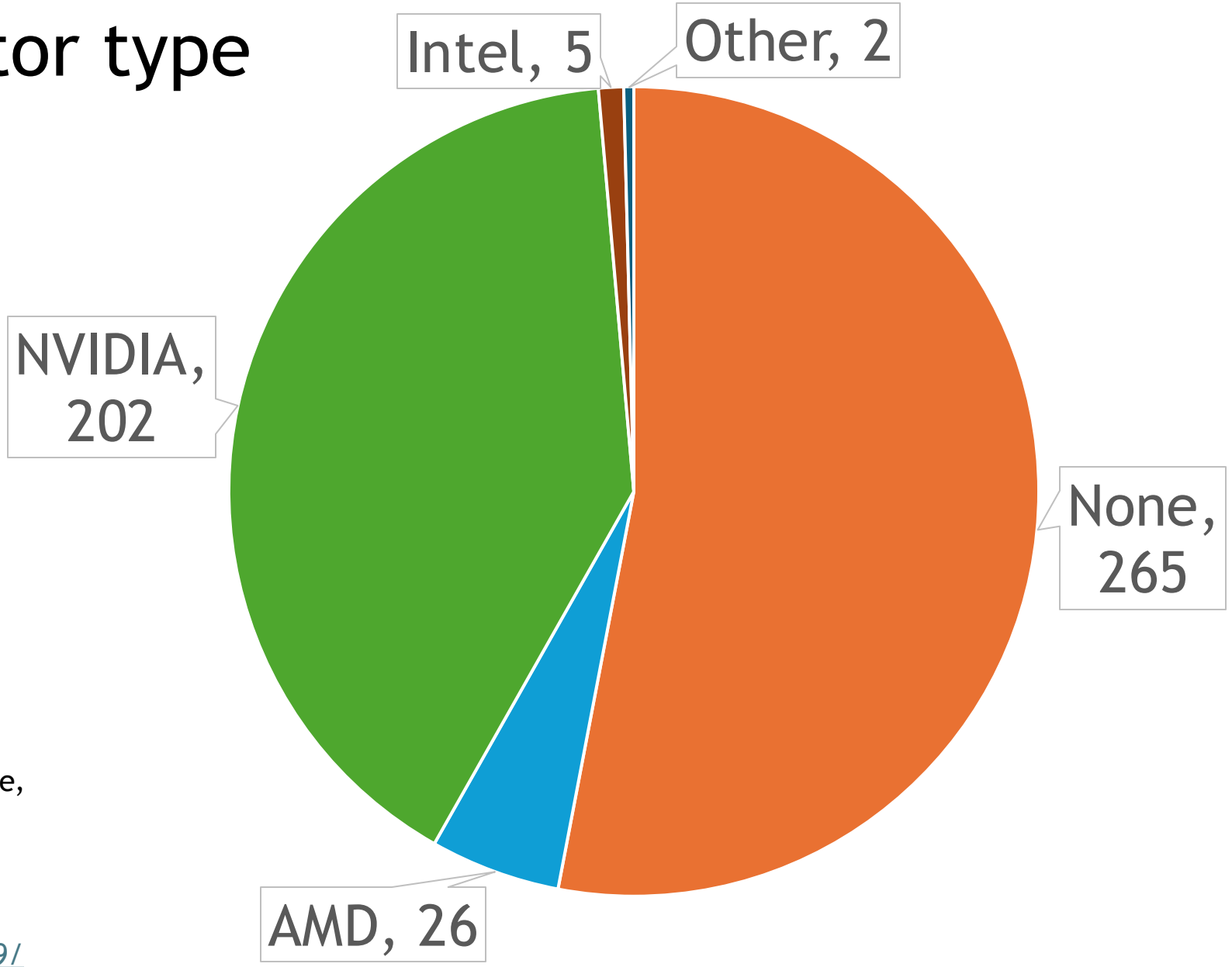
Benchmarking the Three Ps: Performance, Portability, and Productivity

Tom Deakin – University of Bristol

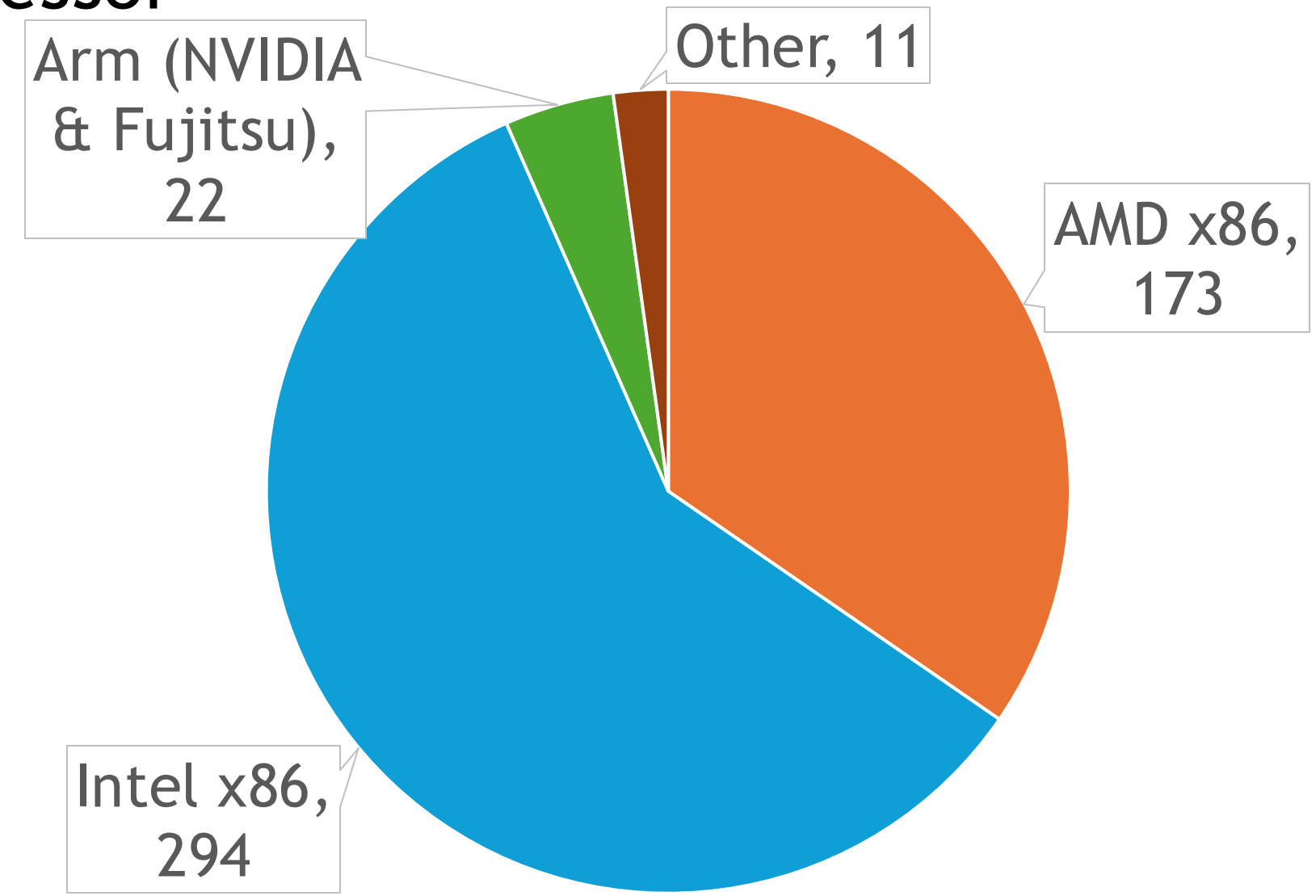

Generated by AI

Accelerator type (count)



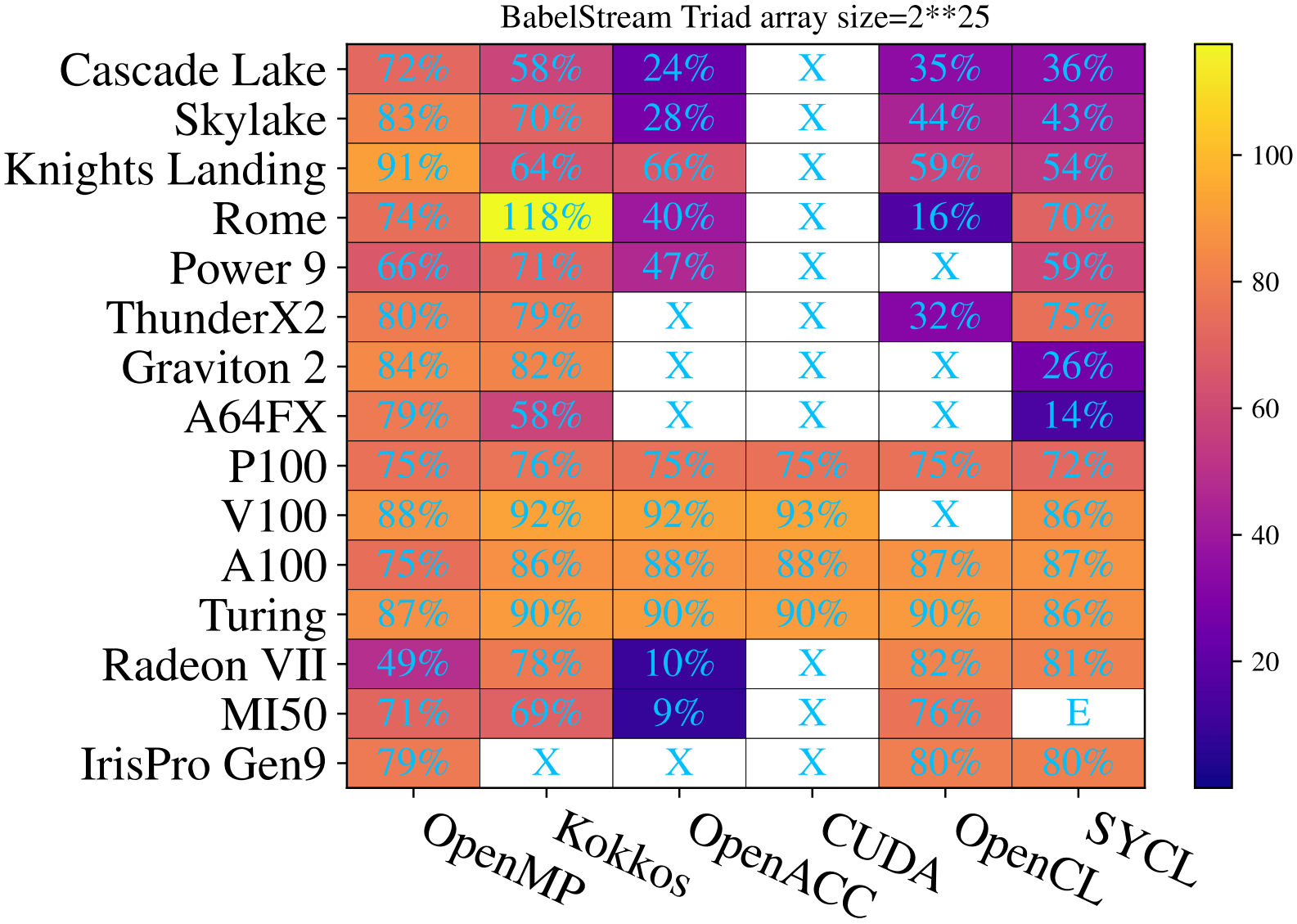
Data: Top500 June 2025
Update from figure shown in Deakin, Cownie, Lin, McIntosh-Smith, Heterogeneous Programming for the Homogeneous Majority
https://doi.org/10.1109/P3HPC56579.2022.00006

Host processor (count)

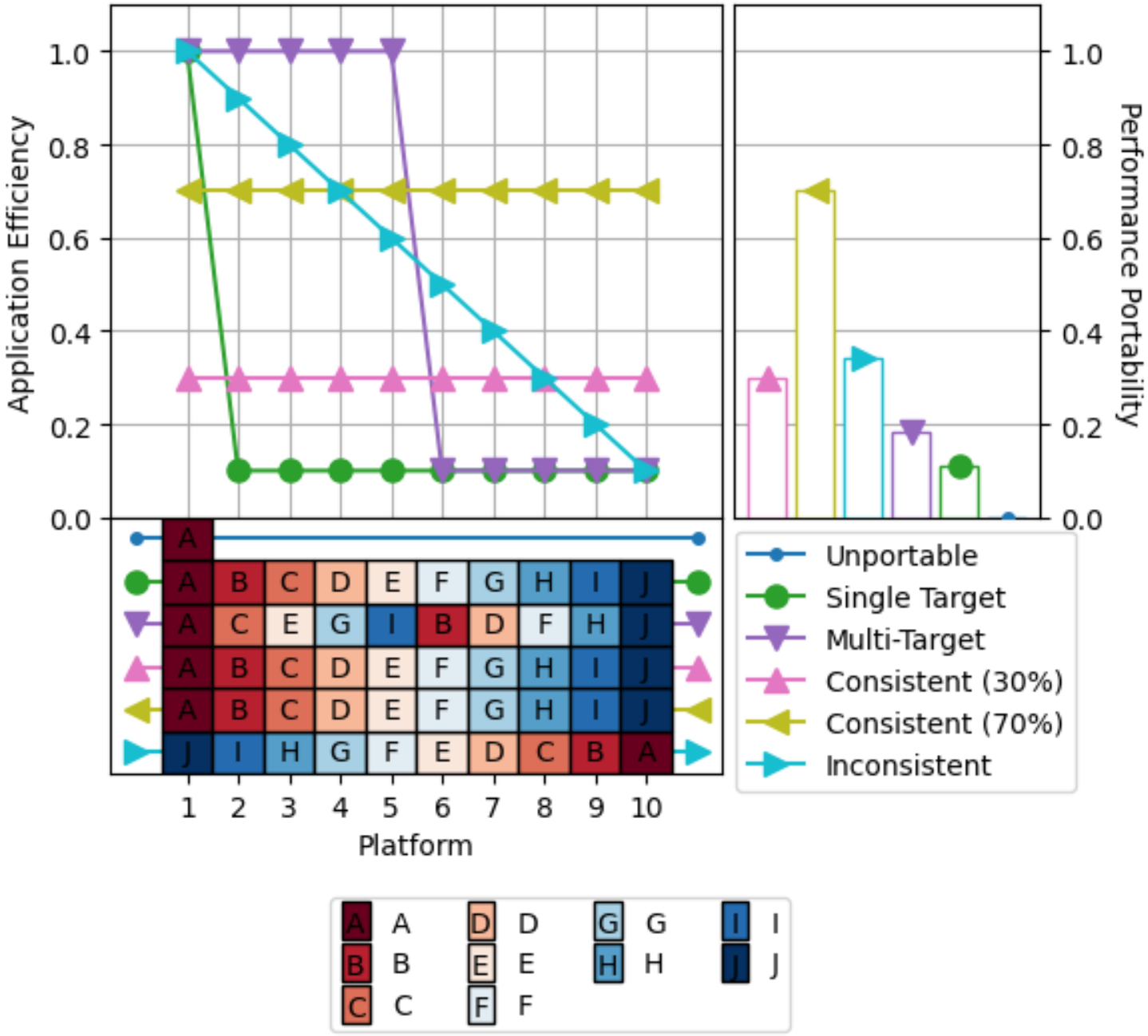


Data: Top500 June 2025

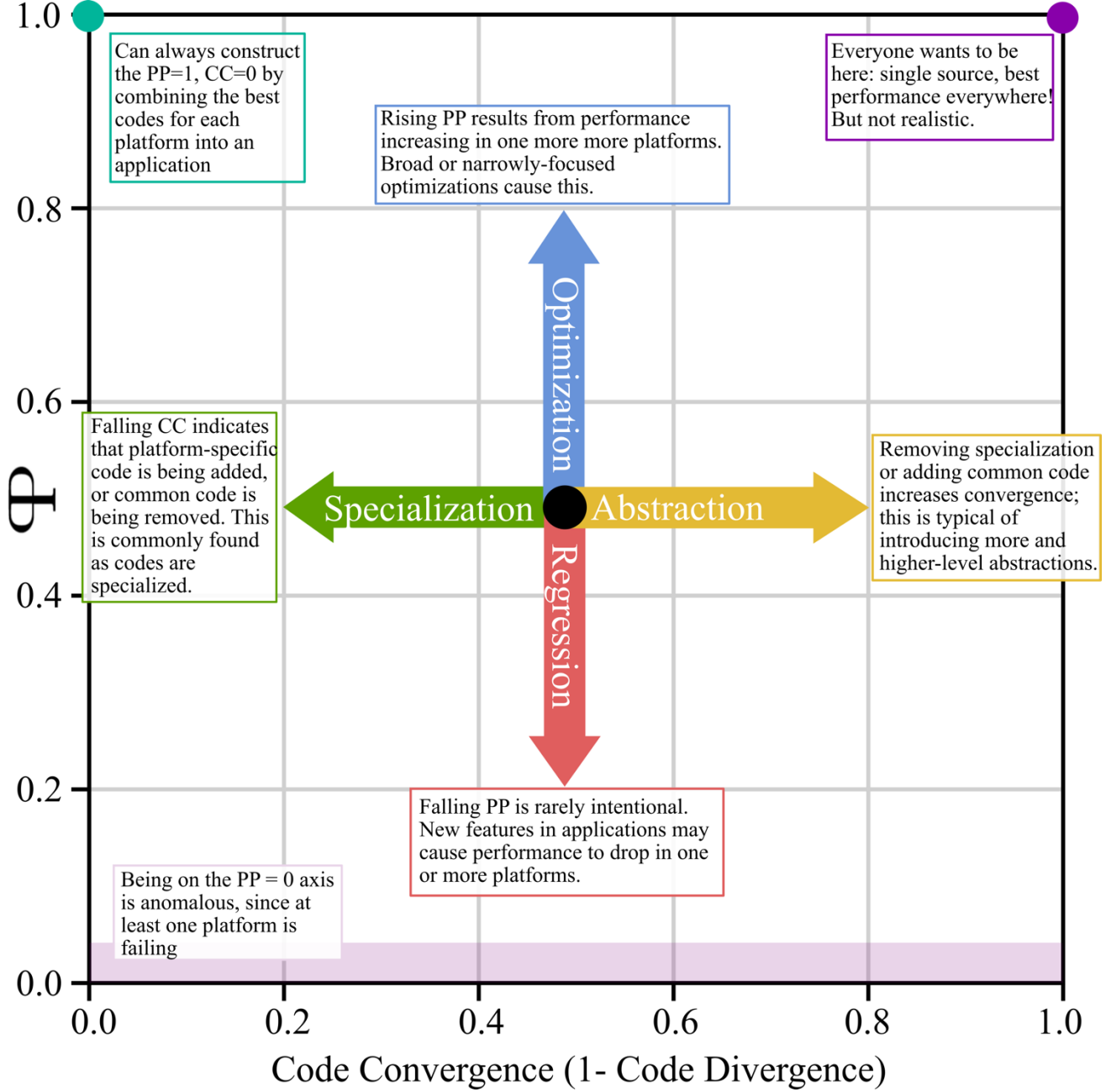
Generated by AI



https://doi.org/10.1109/P3HPC51967.2020.00006



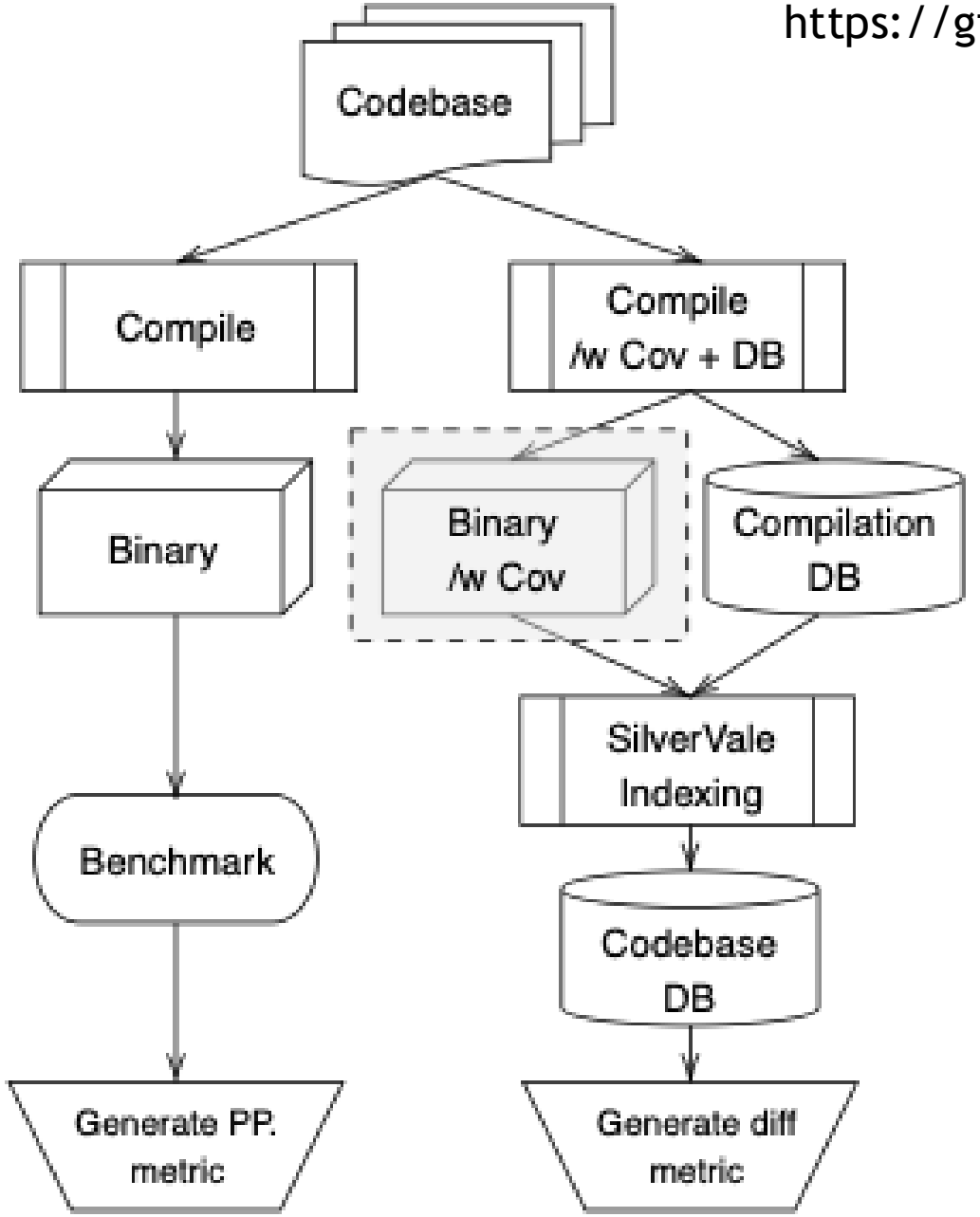
From https://intel.github.io/p3-analysis-library
Based on Pennycook, Sewall, Jacobsen, Deakin, McIntosh-Smith *Navigating Performance, Portability, and Productivity* https://doi.org/10.1109/MCSE.2021.3097276



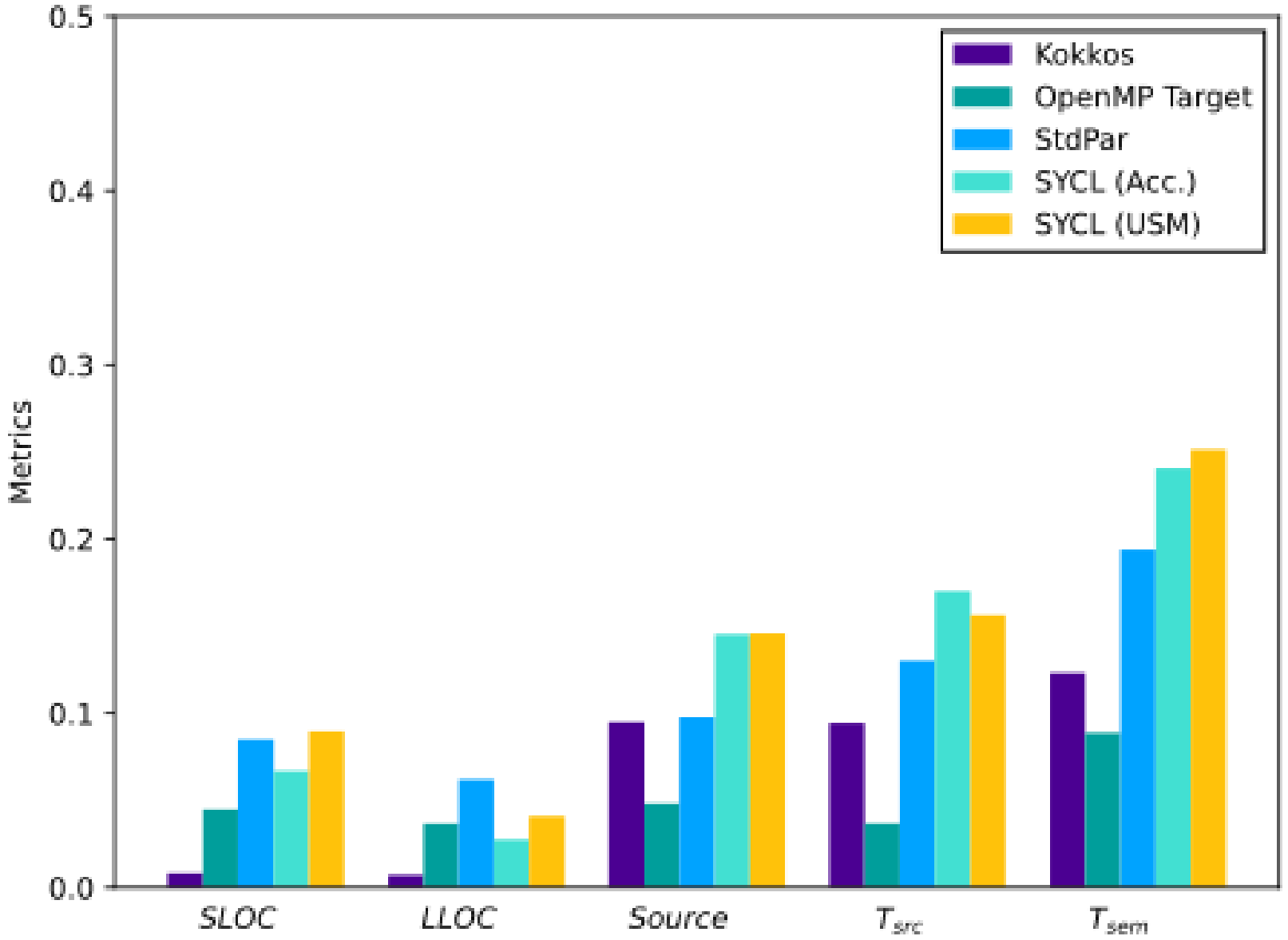
From Pennycook, Sewall, Jacobsen, Deakin, McIntosh-Smith *Navigating Performance, Portability, and Productivity* https://doi.org/10.1109/MCSE.2021.3097276

SilverVale

https://github.com/UoB-HPC/SilverVale



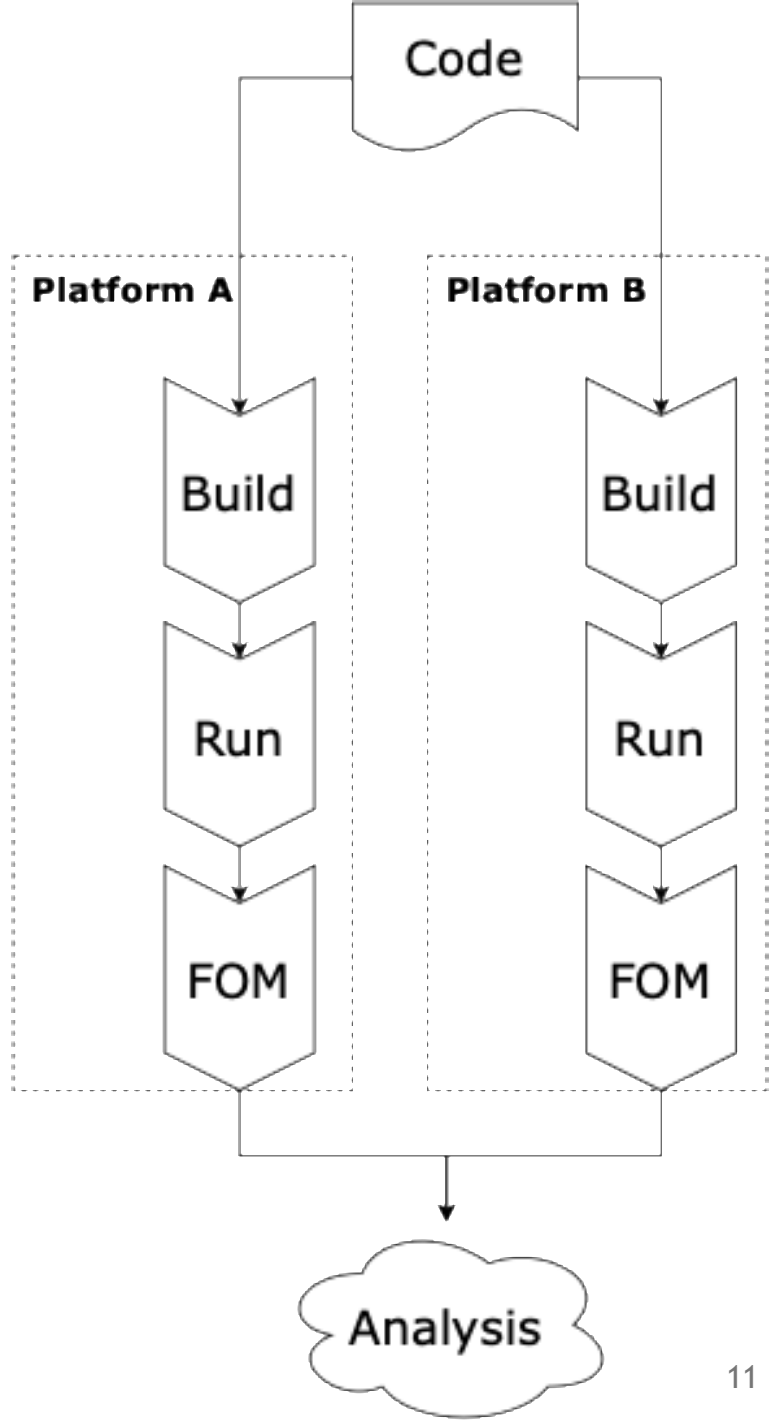
Lin, Deakin, and McIntosh-Smith. A Metric for HPC Programming Model Productivity.P3HPC 2024, https://doi.org/10.1109/SCW63240.2024.00160.



Lin, Deakin, and McIntosh-Smith. A Metric for HPC Programming Model Productivity.P3HPC 2024, https://doi.org/10.1109/SCW63240.2024.00160.

ReFrame

Spack



From Koskela, et al, Principles for Automated and Reproducible Benchmarking, HPCTESTS, 2023 https://doi.org/10.1145/3624062.3624133

Living Benchmarks: Developing benchmarks that will influence future Digital Research Infrastructure investments

Good benchmarks ensure that all target community needs are met

Reaching across UKRI disciplines and communities

https://ukri-bench.github.io

The project is supported by the UKRI Digital Research Infrastructure Programme under grant APP46895.

Some unsolved challenges

- Building *heterogeneous* software is still hard
  - Spack isn't a panacea – still requires per-platform specalisation (variants)
- Tension in automated tooling for running benchmarks
  - Ramble and ReFrame great options for recording the process (reproducing)
  - Tension between platform-specific changes and out of box behaviour
  - Just as hard to set up experiment on a "heterogeneous zoo", but at least it recorded what you did automatically
- How do I know if my benchmark is efficient on that platform?
  - Performance modelling tools insufficient to model program/kernel efficiency
  - E.g., lots of good tools to *measure* Roofline, but still need intuition/analytical model to verify arithmetic intensity is correct

References

Koskela, Christidi, Giordano, Dubrovska, Quinn, Maynard, Case, Olgu, and Deakin. “Principles for Automated and Reproducible Benchmarking.” In First International Workshop on HPC Testing and Evaluation of Systems, Tools, and Software Held in Conduction with Supercomputing (HPCTESTS). IEEE, 2023. https://doi.org/10.1145/3624062.3624133

Deakin, Poenaru, Lin, and McIntosh-Smith. “Tracking Performance Portability on the Yellow Brick Road to Exascale.” In International Workshop on Performance, Portability and Productivity in HPC Held in Conjunction with Supercomputing (P3HPC). IEEE, 2020. https://doi.org/10.1109/P3HPC51967.2020.00006

Deakin, Cownie, Lin, and McIntosh-Smith, “Heterogeneous Programming for the Homogeneous Majority.” In International Workshop on Performance, Portability and Productivity in HPC (P3HPC), 2022. https://doi.org/10.1109/P3HPC56579.2022.00006

Lin, Deakin, and McIntosh-Smith. “A Metric for HPC Programming Model Productivity.” In International Workshop on Performance, Portability and Productivity in HPC Held in Conjunction with Supercomputing (P3HPC). IEEE, 2024. https://doi.org/10.1109/SCW63240.2024.00160

Pennycook, Sewall, Jacobsen, Deakin, and McIntosh-Smith, “Navigating Performance, Portability and Productivity.” Computing in Science and Engineering, 2021. https://doi.org/10.1109/MCSE.2021.3097276