PyPOP — An interactive tool for performance assessment

Phil Tooley (phil.tooley@nag.co.uk) — NAG

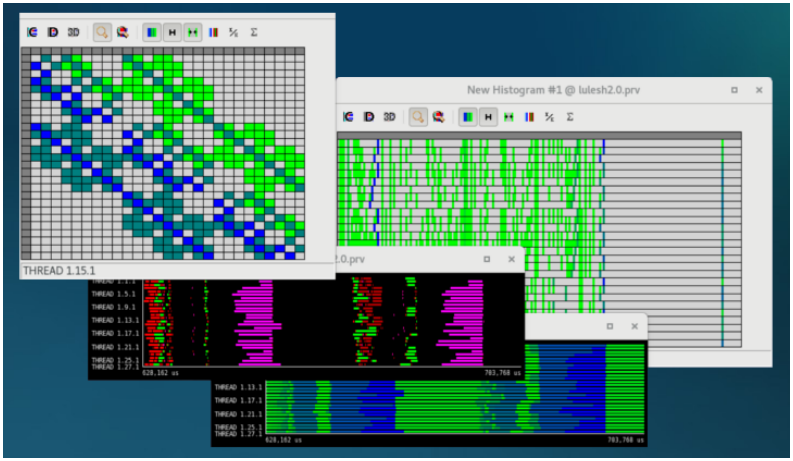
EU H2020 Centre of Excellence

Grant Agreement No 824080

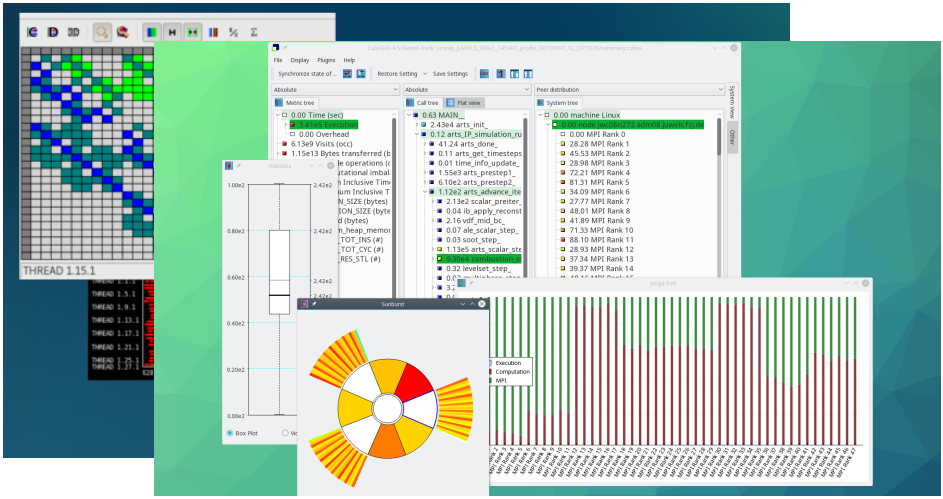
Why another new tool?

Why does POP need another profiling tool?

Why does POP need another profiling tool?



Why does POP need another profiling tool?



A tool for the POP Methodology

POP Metrics Goals

- Small set of quantitative measurements
- Easy to calculate and understand performance
- Quickly determine what to investigate further

Number of Processes	1	2	4	8	16
Global Efficiency	1.00	0.95	0.90	0.78	0.58
↪ Parallel Efficiency	1.00	0.99	0.97	0.96	0.95
↪ Load balance	1.00	1.00	0.99	0.98	0.99
↪ MPI Communication Efficiency	1.00	1.00	0.99	0.97	0.96
↪ MPI Transfer Efficiency	1.00	1.00	0.99	0.99	0.99
↪ MPI Serialisation Efficiency	1.00	1.00	0.99	0.98	0.97
↪ Computation Scaling	1.00	0.96	0.92	0.81	0.61
↪ Instruction Scaling	1.00	0.97	0.96	0.94	0.91
↪ IPC Scaling	1.00	1.00	1.00	0.94	0.82
↪ Frequency Scaling	1.00	0.99	0.97	0.92	0.82

A lightweight tool for the POP Methodology

PyPOP Goals

1. Easy to calculate the POP Metrics
2. Easy to visualise and explore the metrics
3. Easy to share and annotate results
4. Tools to help with further analysis

A lightweight tool for the POP Methodology

PyPOP Goals

1. Easy to calculate the POP Metrics
2. Easy to visualise and explore the metrics
3. Easy to share and annotate results
4. Tools to help with further analysis

PyPOP is *not* another profiler!

- Use existing profilers (Extrae)
- Focus on ease of use

PyPOP in Action

1. Calculating the POP Metrics

GUI Tool and CLI Tool for Metrics and Scaling

1. Collect traces with supported tool (Extrae)
2. (Optional) preprocess traces as a batch job
3. Run GUI/CLI analysis tool to calculate metrics and scaling

2. Visualising and Exploring the POP Metrics

GUI Tool and CLI Tool for Metrics and Scaling

1. Collect traces with supported tool (Extrae)
2. (Optional) preprocess traces as a batch job
3. Run GUI/CLI analysis tool to calculate metrics and scaling

Plotting Features

- Scaling plots and POP Metrics Tables
- Interactive GUI elements in notebook interface
- CSV and PNG output from CLI for remote machines

3. Sharing and Annotating the Results

Sharing Features

1. Create static notebook from GUI
2. Add descriptions and discussion
3. Convert to PDF and share

4. Extra tools — OpenMP Region Explorer

Easy to write extensions

- Python API with Numpy, Pandas, Bokeh internals
- Easy to extend for custom functionality
- E.g OpenMP region explorer

Summary

Summary

A tool for efficient performance analysis workflows

- Quickly analyse traces and compute POP metrics
- Produce publication-ready metric tables and scaling graphs
- Output fully-formatted PDF Reports
- Extensible framework using standard data-science tools

PyPOP Github

- https://github.com/numericalalgorithmsgroup/pypop