Introduction

NAG are delighted to be part of the EU group called POP (Performance Optimisation and Productivity) that is helping to improve the performance of software. In brief POP offers to analyse software and recommend improvements with a focus on HPC and parallelization. This service is free of charge to EU organisations.

Methodology

Efficiency metrics give an overview of how well the parallelization of the applications works and how efficient the hardware is used. Each efficiency metric investigates one source of common inefficiency in a parallel program.

The metrics are organized in a hierarchy which allows you to drill down from the top level Global Efficiency to specific inefficiencies in a program, they give a detailed overview of the parallel performance of an application in a very condensed form.

- **Global Efficiency** - Overall performance
  - **Parallel Efficiency** - Efficiency of parallelization strategy
    - **Load Balance Efficiency** - Distribution of work
    - **Communication Efficiency**
      - **Serialization Efficiency** - Dependencies between processes
      - **Transfer Efficiency** - Effect of data transfer
    - **Computational Efficiency** - Scaling of computational load
      - **IPC Scaling** - Implicates resource contention
      - **Instruction Scaling** - Increase in computational work

Tools

Integral to the POP project is the use of performance analysis tools, some developed by members of the POP consortium.

**Barcelona tools**

Include **Paraver**, a trace-based performance analyser with great flexibility to explore and extract information. Including timelines that graphically display the evolution of the application, and tables that provide statistical information.

![Figure 3: Paraver timeline](image)

**Jülich tools**

Include **Scalasca** which characterises parallel execution inefficiencies and detects the best candidates for optimisation.

![Figure 4: Cube view of Scalasca data from MPI test program](image)

**Intel® tools**

Include **VTune** which is a powerful tool for profiling code written in a variety of languages including C/C++, Fortran and Python. It can be used on parallel code that uses paradigms such as OpenMP, MPI and Intel® Thread Building Blocks (TBB), as well as on serial code.

![Figure 5: VTune analysis of Cheby code.](image)

Breadth of Work

Organisations from a wide range of application areas have used the services:

- Computer aided engineering (CAE)
  - Finite element analysis (FEA)
  - Computational fluid dynamics (CFD)
- Earth sciences
- Neural networks
- Materials & Chemistry modelling
- Electronic structure calculations
- Health

Along with many languages and parallelization approaches:

Acknowledgements

The other POP project partners: Barcelona Supercomputing Center, Juelich Supercomputing Center, RWTH Aachen, HLRS Stuttgart and Teratec.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 676553.