The POP Centre of Excellence - Improving Parallel Codes

Lucas, C.[1], Tooley, P.[2]

[1]Craig Lucas, Senior Technical Consultant, The Numerical Algorithms Group. craig.lucas@nag.co.uk
[2]Philip Tooley, HPC Applications Analyst, The Numerical Algorithms Group. phil.tooley@nag.co.uk

1. Introduction

The Performance Optimisation and Productivity (POP) Centre of Excellence [1] is funded through Horizon 2020, like CompBioMed, and is made up of eight partners across Europe [2]. Our remit is to improve the performance of both academic and commercial parallel codes. Working with developers and users we promote a methodology for understanding a code's performance which helps us go on to improve it.

We will present the importance of code analysis and the systematic approach we use. Measuring application performance often results in large amounts of data that is difficult to handle or interpret beyond some simple first observations. The methodology used within the POP CoE provides a quantitative way of measuring the relative impact of the different factors inherent in parallelisation, each metric reflecting a common cause of inefficiency. This provides the knowledge necessary to decide the best course of action to get performance via reproducible and comparable measurements of the performance.

We will highlight some of the open source profiling tools used within the project as well as some code improvements made so far. Details will be given as to how the POP service can be utilized.

2. Examples of Performance Analysis and Improvements

Figure 1 plots the poor load balance of I/O for an Electron-Phonon Wannier (EPW) materials science DFT code. Additional to this, POP analysts also identified poor load balancing of computation. Improvements were made to address both issues, enabling simulations to scale to a previously impractical 1920 MPI ranks and achieve 86% efficiency with 960 MPI ranks.

In Figure 2 we show the before and after time spent in a data analysis application. Here, the turquoise segments are idle time. Several algorithmic improvements were made to the code by POP as well as the addition of nested OpenMP parallelization. OpenMP tasks were used improve the load balancing and reduce the runtime by up to 44%.

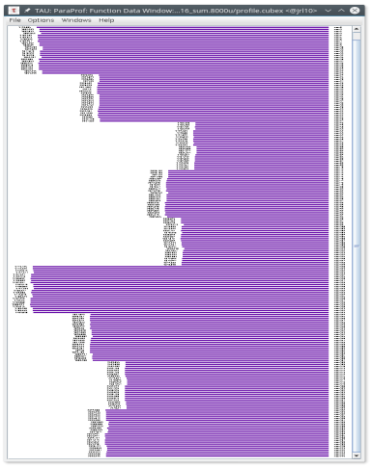

Figure 1 Histogram showing varied writing time for a materials science code, contributing to poor load balance.

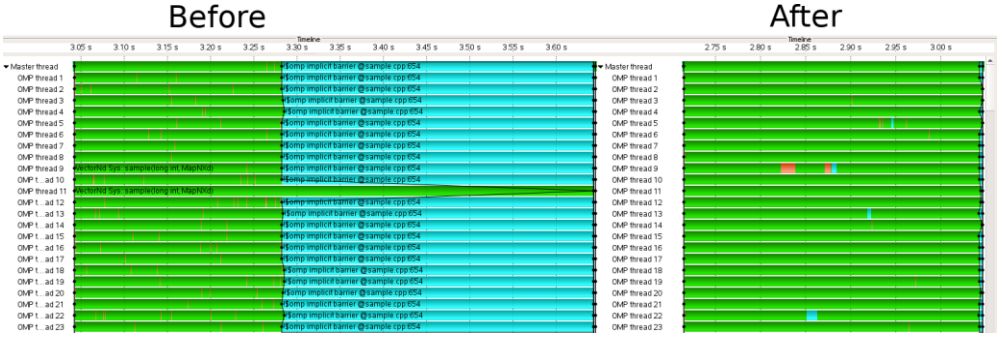

Figure 2 A data analysis code where most of the idle time, shown turquoise, has been eliminated to achieve a well load balanced application.

3. References

[1] The POP Website. https://pop-coe.eu/

[2] The POP Partners are Barcelona Supercomputer Center, High Performance Computing Center Stuttgart, IT4Innovations National Supercomputing Center, Jülich Supercomputing Centre (JSC), he Numerical Algorithms Group Ltd, Rheinisch-Westfälische Technische Hochschule Aachen, Teratec, LI-PaRAD Laboratory UVSQ. https://pop-coe.eu/partners