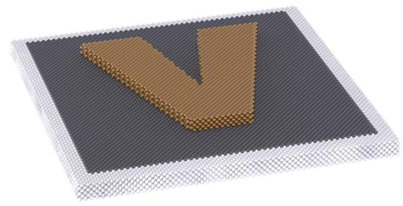
Example of POP Parallel Performance Audit

Sally Bridgwater, Numerical Algorithms Group Ltd, Oxford, UK
Richard Evans, Roy Chantrell, The University of York, York, UK



Introduction

The *Performance Optimisation and Productivity Centre of Excellence in Computing Applications* (POP) performed an Audit on the VAMPIRE magnetic simulation package from The University of York, as described in this poster.

The audit investigates the general performance of a code, the following four key areas provide a well-rounded description of the performance and potential for optimisation so developers can focus their efforts effectively to optimise performance.

This service is free to organisations within the EU.

Serial Performance

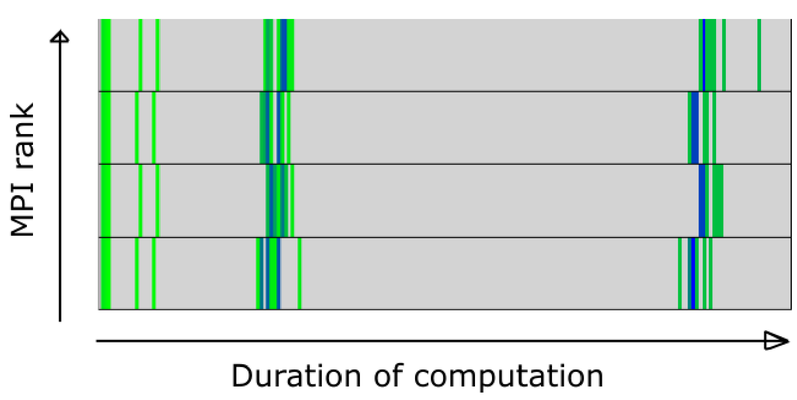
Serial performance is investigated via the instructions per cycle (IPC) metric and hardware counters in order to reduce computation time.

The VAMPIRE code spends most of its time ($\sim$ 70%) in just three functions. Use of hardware counters showed the cause to be algorithmic issues with divisions, low cache reuse and minimal vectorization.

Load Balance

Poor load balance leads to idle processors that waste computational capability.

In the VAMPIRE code, the load balance across the entire program is very good. However, there is slight internal imbalance shown by the offset in duration for different MPI ranks in the figure below.



Efficiency

The achieved efficiencies of the code for communication and balance across MPI ranks are summarised, values above 85% indicate good performance.

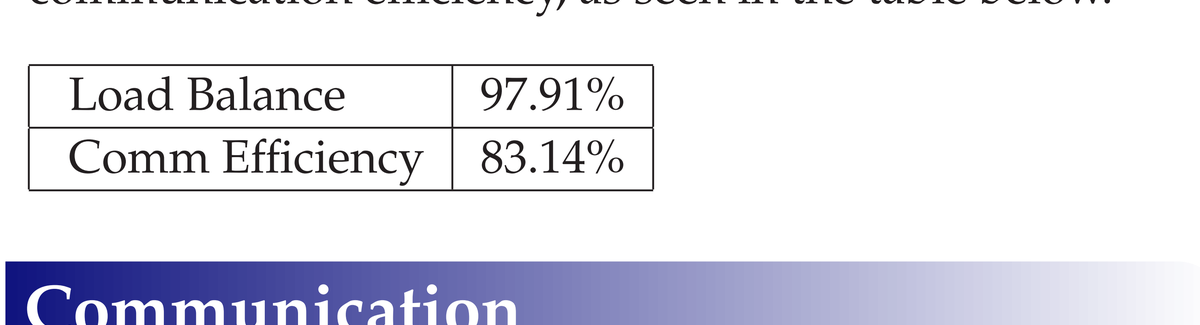
The VAMPIRE code was found to have very good load balance but possible room for improvement in communication efficiency, as seen in the table below.

Load Balance	97.91%
Comm Efficiency	83.14%

Communication

Efficient communication will improve performance and scalability of the code, allowing full utilisation of HPC resources.

In VAMPIRE the slight imbalance between ranks was found to be due to the ordering of the send and receives, as illustrated below.



Conclusion

Recommendations for the VAMPIRE code described how to improve the cache re-use, vectorization and ordering of the MPI send and receives.

*"The audit of the VAMPIRE code has been **extremely helpful** in identifying the hot spots and specific areas to focus on performance improvements. Preliminary results suggest this may give a **factor of 2 performance improvement** on modern CPUs. I would **highly recommend** the service for the speed and usefulness of the audit."*
- **Richard Evans**, VAMPIRE developer

https://pop-coe.eu

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