What is Performance Portability?

- Performance Portability: the ability for a single-source application to run on a range of hardware platforms while maintaining good performance.

- OpenMP target offload (OMPT), OpenACC (ACC), Kokkos, RAJA, SYCL and HIP are programming models providing portable abstractions.

Methodology for Evaluating Perf. Portability

- We surveyed available proxy applications and benchmarks, and selected those with the most available implementations. This poster focuses on two memory-bound proxy applications: BabelStream and XSBench.

Abstract

Maintaining a single codebase that can achieve good performance on a range of accelerator-based supercomputing platforms is of extremely high value for productive scientific application development. However, the large quantity of programming models that claim to provide performance portability leaves developers with a complex choice when picking a model to use. In order to better understand the current state of performance portable programming models, this project evaluates seven of the most popular programming models using two memory-bound proxy applications on two leadership-class supercomputers, Summit and Perlmutter. These results provide a useful evaluation of how well each programming model provides performance portability in real-world usage for memory-bound applications.

Comparative Evaluation on Summit (V100) and Perlmutter (A100)

- Performance metrics are measured in terms of runtime in XSBench, so lower is better.
- Kokkos outperforms even CUDA on both systems.
- OpenMP/ACC lag far behind on Perlmutter, but only moderately slower on Summit.
- HIP performs poorly on Summit.
- SYCL performs competitively on Summit but not Perlmutter.
- Higher variability across models on Perlmutter.

Performance Portability Metric and Discussion

- Performance portability metric from Penykoor et al. [3] is defined as the harmonic mean of performance efficiency.
- We define performance efficiency as application efficiency, the performance of the app implementation in a model divided by peak performance achieved across all implementations on that platform.

Conclusion and Future Work

- Results set expectations for developers looking to select a programming model for a memory-bound abstraction, and for those porting their application from Summit V100s to Perlmutter A100s.
- Summit and Perlmutter both use NVIDIA GPUs – moving to Frontier (AMD) and Aurora (Intel) will provide even greater challenge.

Acknowledgements

- OLCF Summit: IBM Power 9 CPU and NVIDIA V100 GPU
- NERSC Perlmutter: AMD EPYC CPU and NVIDIA A100 GPU

See more results here

References

