Evaluating Performance Portability of GPU Programming Models

Joshua H. Davis, Pranav Sivaraman, Isaac Minn, Abhinav Bhatele
Department of Computer Science, University of Maryland

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 available 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.

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

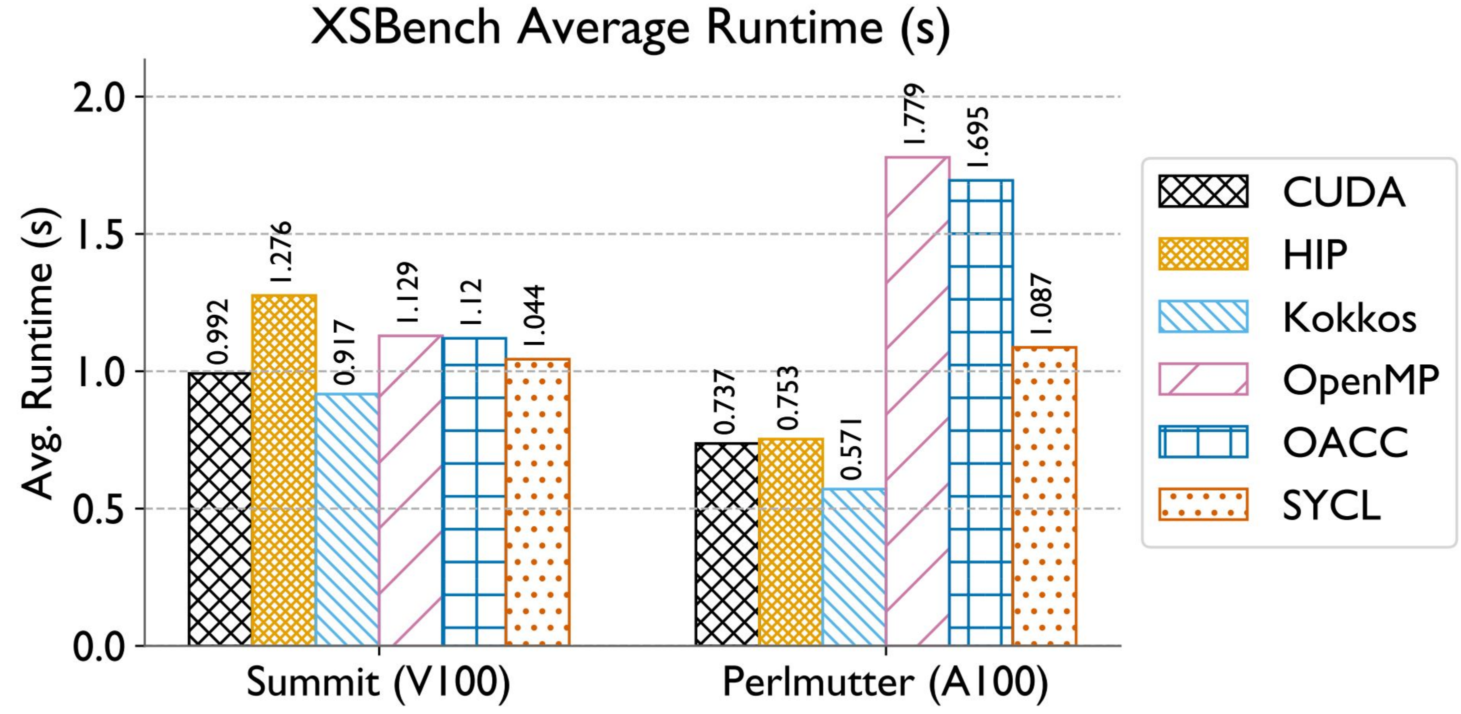
Directive-based — OpenMP, OpenACC
Library-based — kokkos, RAJA
Language Extension — AMD ROCm, SYCL, NVIDIA CUDA

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 codes.
- **XSBench [1]**: memory-bound proxy app from OpenMC (Monte Carlo), evaluated with the `large` problem size (355 isotopes, 11303 grid points)
  - We implemented a new Kokkos port of XSBench for this effort
- **BabelStream [2]**: a memory bandwidth benchmark. We evaluate the dot, triad, and copy kernels for 800 iterations each
- Evaluation platforms:
  - OLCF Summit: IBM Power 9 CPU and NVIDIA **V100** GPU
  - NERSC Perlmutter: AMD EPYC CPU and NVIDIA **A100** GPU

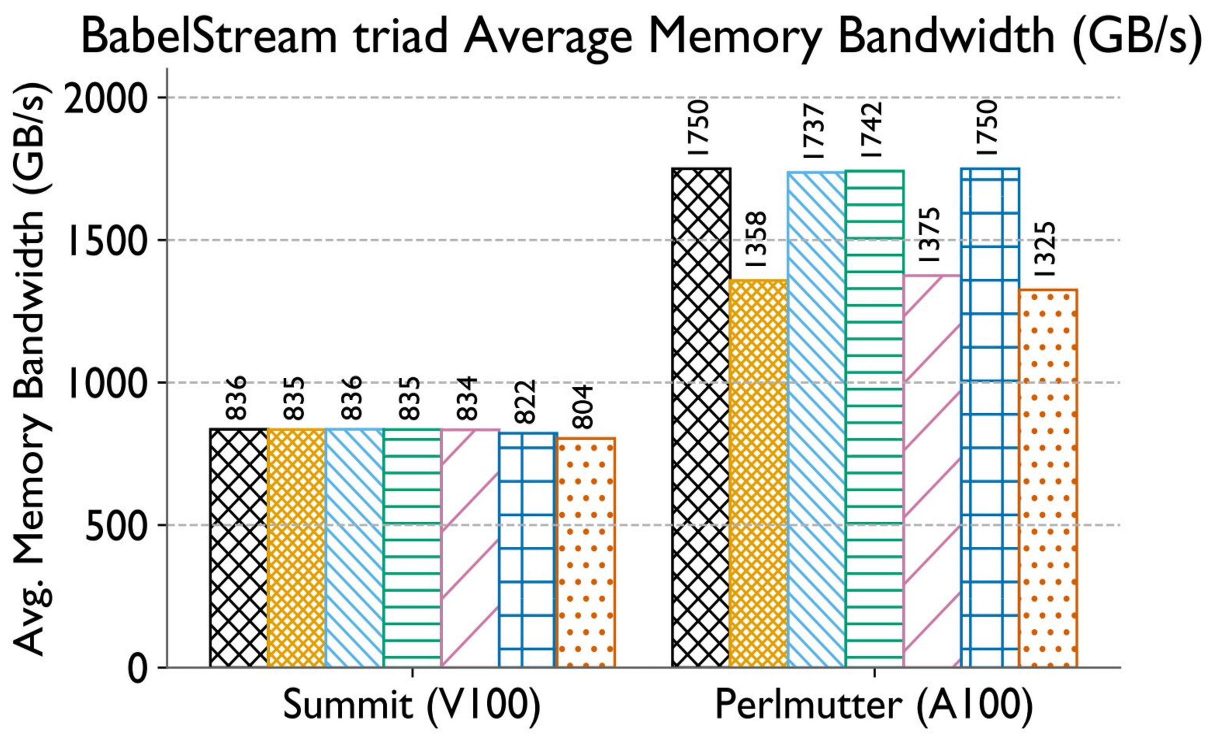
Comparative Evaluation on Summit (V100) and Perlmutter (A100)

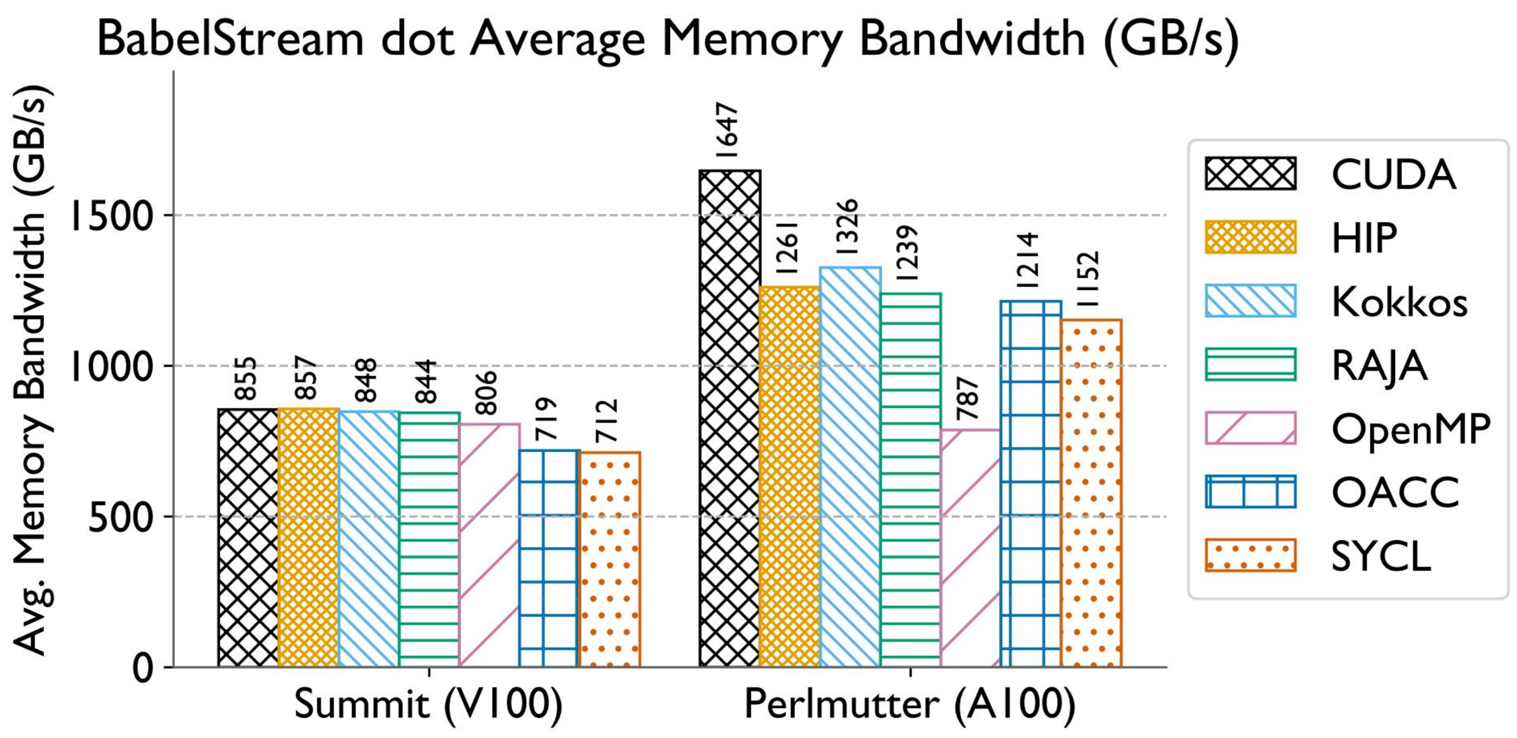
- Performance is measured in terms of runtime in XSBench, so **lower is better**
- Kokkos outperforms even CUDA on both systems
- OpenMP/ACC lag far behind on Perlmutter, but only moderately slower on Summit
- HIP performs poorly on Summit
- SYCL performans competitively on Summit but not Perlmutter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



- BabelStream performance is measured in terms of memory transfer bandwidth, so **higher is better**
- All models struggle with dot (reduction) on Perlmutter, OpenMP is a dramatic low outlier
- SYCL is the worst performer in all other cases
- All models deliver near-CUDA performance on Summit for triad
- Kokkos and RAJA both able to perform near CUDA on Perlmutter triad and Summit dot as well
- OpenACC near the bottom for Summit but matches CUDA on Perlmutter triad
- Again, higher overall variability on Perlmutter

See more results here





Performance Portability Metric and Discussion

Performance Portability Metric (Pennycook et al., [3])

$$\Phi(a,p,H) = \begin{cases} \dfrac{|H|}{\sum_{i \in H} \dfrac{1}{e_i(a,p)}} & \text{if } i \text{ is supported } \forall i \in H \\ 0 & \text{otherwise} \end{cases}$$

- Performance portability metric from Pennycook 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

Application	OMPT	ACC	Kokkos	RAJA	SYCL	HIP	CUDA
XSBench	0.46	0.48	1.00	0.84	0.66	0.74	0.84
BS-Copy	0.88	0.98	0.99	1.00	0.85	0.88	1.00
BS-Triad	0.88	0.99	1.00	1.00	0.85	0.87	1.00
BS-Dot	0.63	0.78	0.89	0.85	0.76	0.87	1.00

- Kokkos, CUDA, and HIP achieve the best performance portability, OpenMP and OpenACC are the worst
- The much larger and more complex kernel in XSBench and the reduction operation in BabelSteam-dot lead to worse performance portability for most models

Conclusion and Future Work

- Results set expectations for developers looking to select a programming model for a memory-bound application, 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.
- XSBench RAJA, CloverLeaf, su3_bench, and Frontier results are available and will be shown at Best Research Poster session.
- Continuing to analyze the performance of additional applications and programming models

Acknowledgements

This material is based upon work supported in part by the National Science Foundation Graduate Research Fellowship under Grant No. 1650114. This research used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725. This research used resources of the National Energy Research Scientific Computing Center (NERSC), a U.S. Department of Energy Office of Science User Facility located at Lawrence Berkeley National Laboratory, operated under Contract No. DE-AC02-05CH11231 using NERSC award m2404 for 2023.

References

[1] John R Tramm, Andrew R Siegel, Tanzima Islam, and Martin Schulz. 2014. XSBench-the development and verification of a performance abstraction for Monte Carlo reactor analysis. *PHYSOR.* (2014).
[2] Tom Deakin, James Price, Matt Martineau, and Simon McIntosh-Smith. 2018. Evaluating Attainable Memory Bandwidth of Parallel Programming Models via BabelStream. *Int. J. Comput. Sci. Eng.* 17, 3 (Jan 2018), 247–262.
[3] Simon J Pennycook, Jason D Sewall, and Victor W Lee. 2016. A metric for performance portability. In *Proceedings of the 7th International Workshop in Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems.*