Porting HPCG to Kokkos for Enhanced Portability and Performance

Muhammad Rizwan* and Jaeyoung Choi

School of Computer Science and Engineering, Soongsil University, Seoul, Republic of Korea

Corresponding author (Electronic mail: mrizwan@soongsil.ac.kr)

The High-Performance Conjugate Gradient (HPCG) benchmark [1] has emerged as a counterpart to High Performance Linpack (HPL) to evaluate system performance by utilizing sparse matrix computations. HPL concentrates on the peak computational performance measure, whereas HPCG reflects the realistic performance measure and stresses the coordination of all parts of the system, such as memory bandwidth, computation power, communication, and the interconnect network of modern supercomputers. However, its performance is constrained by the sequential nature of the Symmetric Gauss-Seidel (SymGS) and Sparse Matrix-Vector Multiplication (SpMV) key kernels. The efforts required to optimize HPCG have primarily focused on increasing parallelism to overcome its inherent performance bottlenecks for the performance improvement over various architectures. The Intel optimized variant offers improvements, but the current Kokkos-based implementation KHPCG has concerns regarding its scalability and the validity of the results. In this work, we modify an optimized variant of HPCG and port it to the Kokkos framework by modifying the loop structures with Kokkos parallel dispatch functors or lambdas. Our current results show a slight drop in performance compared to the optimized version, but a significant improvement over the native HPCG and KHPCG implementations. In the future, we intend to extend this work using the Kokkos data structure abstractions to improve portability across CPU and GPU architectures, so that it can be widely applicable in future heterogeneous architectures.

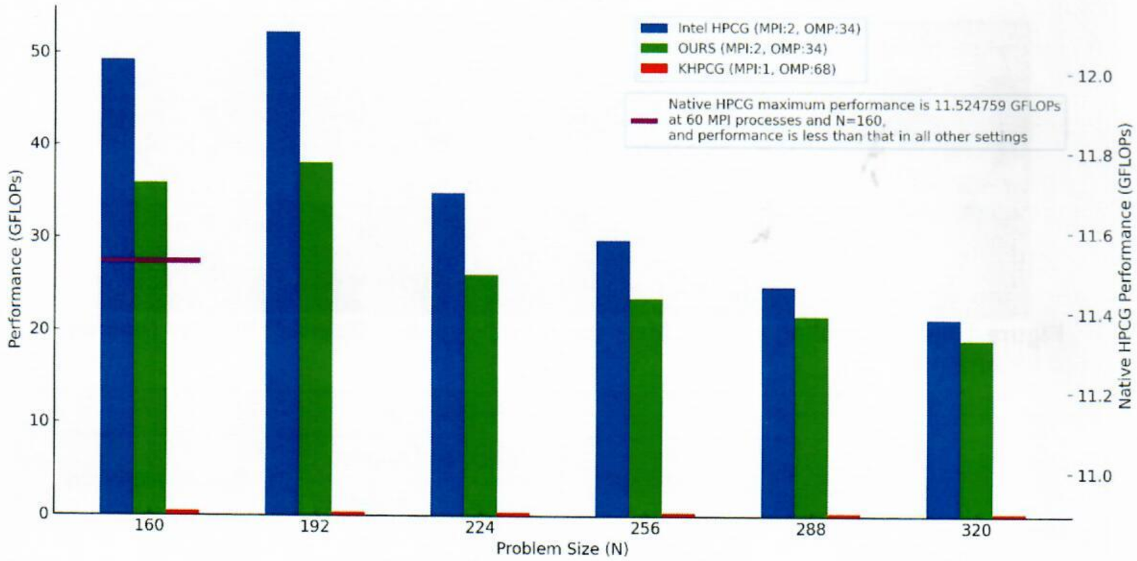

Figure 1 Performance comparison of Intel HPCG, OURS, and KHPCG for various problem sizes (N), conducted on a single-node Intel KNL. The purple horizontal line shows the maximum Native HPCG performance of 11.524759 GFLOPs at 60 MPI processes for N=160. All other settings perform below this maximum. The legend also lists MPI and OpenMP configurations for each variant.

Acknowledgments This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT), and this work was also supported by the Korean National Supercomputing Center (KSC) with supercomputing resources.

References

[1] J. Dongarra et al., "HPCG technical specification," Sandia National Laboratories, Sandia Report SAND2013-8752, (2013).