OOKAMI PROJECT APPLICATION

DATE: 03/11/2021

Project Title: Optimizing Stencil Applications with Bricks on A64FX

Usage — Testbed

Principal Investigator: Mary Hall

- University/Company/Institute: *University of Utah*
- Mailing address including country:

School of Computing MEB 3466
University of Utah
Salt Lake City, UT 84112
USA

- Phone number: *(801) 585-1039*
- Email: *mhall@cs.utah.edu*

Initial Users

Mary Hall, University of Utah, mhall@cs.utah.edu
Tuowen Zhao, University of Utah, ztuowen@cs.utah.edu
Samuel Williams, Lawrence Berkeley National Laboratory, swwilliams@lbl.gov
Hans Johansen, Lawrence Berkeley National Laboratory, hjohansen@lbl.gov

Usage description

Brick is a data layout and code generation framework to enable the development of performance-portable stencil applications. We have developed vector code generation strategies from single source code on Intel CPUs, Intel Knights Landing, ARM CPUs, NVIDIA GPUs, and Intel GPUs. We are able to achieve best-in-class performance on many of these platforms and significantly improve communication performance by optimizing data movement.

Stencil applications' performance is intimately influenced by the memory, compute, and communication capability of the platform. We will do performance characterization for this supercomputer including roofline and network latency/throughput analysis. These characteristics will guide a performance portability study.

In this project, we intend to enable vector code generation for A64FX processors with SVE instructions in our code generator. We also intend to optimize the vertical and horizontal data movement on this platform using the brick data layout. We will work with a set of benchmark stencil kernels we have established for other hardware platforms to identify optimization strategies unique to this platform.

Computational Resources

- Total node hours per year: 5,000 node-hour
- Size (nodes) and duration (hours) for a typical batch job:
    - Size: 1 to 64 nodes
    - Duration: 1 hour
- Disk space (home, project, scratch): 200 GB

Personnel Resources

None

Required software

C++ compiler, OpenMP, MPI, Python 3.
Profiler if available.

If your research is supported by US federal agencies

This research was supported by the Exascale Computing Project (17-SC-20-SC), a joint project of the U.S. Department of Energy's Office of Science and National Nuclear Security Administration.

References

1. T. Zhao, S. Williams, M. Hall and H. Johansen, "Delivering Performance-Portable Stencil Computations on CPUs and GPUs Using Bricks," 2018 IEEE/ACM International Workshop on Performance, Portability and Productivity in HPC (P3HPC), Dallas, TX, USA, 2018, pp. 59--70.
2. T. Zhao, P. Basu, S. Williams, M. Hall and H. Johansen, "Exploiting reuse and vectorization in blocked stencil computations on CPUs and GPUs." In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC '19), ACM, Article 52, 1--44.
3. T. Zhao, M. Hall, H. Johansen, and S. Williams. 2021. Improving communication by optimizing on-node data movement with data layout. In Proceedings of the 26th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP '21). Association for Computing Machinery, New York, NY, USA, 304–317.