Title

Just Write Fortran: Experiences with a Language-Based Alternative to MPI+X

Authors

Rouson, Damian
Rasmussen, Katherine
Dibba, Baboucarr
et al.

Publication Date

2024-11-17

DOI

10.25344/S4H88D

Peer reviewed

Just Write Fortran: Experiences with a Language-Based Alternative to MPI+X

Baboucarr Dibba, Katherine Rasmussen, Brad Richardson, Damian Rouson, David Torres, and Yunhao Zhang
Computer Languages and Systems Software (CLaSS) Group,
Lawrence Berkeley National Laboratory, Berkeley, California, USA
{bdibba,krasmussen,brad.richardson,rouson,davytorres,yzhang22}@lbl.gov

Ethan Gutmann
Research Applications Laboratory
National Center for Atmospheric Research, Boulder, Colorado, USA
gutmann@ucar.edu

Kareem Ergawy and Michael Klemm
Advanced Microdevices, Inc.
Munich, Germany
{michael.klemm,kareem.ergawy}@amd.com

Sameer Shende
Performance Research Laboratory, OACISS
University of Oregon, Eugene, Oregon
sameer@cs.uoregon.edu

***Index Terms*—Coarray Fortran, parallel programming, deep learning, high-performance computing, climate modeling.**

In a 2008 paper entitled "Parallel programming: can we PLEASE get it right this time?", Mattson et al. [1] wrote, "With few exceptions, only graduate students and other strange people write parallel software." Parallel programming had already started becoming more widespread in the research community with the 1995 publication on the first distributed-memory Beowulf clusters comprised of networked commodity personal computers [2]. Shared-memory parallelism proliferated in the mid-2000s when the multicore processors first proposed a decade prior [3] reached commodity status contemporaneously with the advent of general-purpose computation on graphics processing units (GPGPUs) [4]. With these hardware trends democratizing parallel computing, the timeliness of the 1996 Message Passing Interface (MPI) specification [5] and the 1997 OpenMP specification explain the widespread use of programming models defined outside of programming languages. But it no longer has to be this way!

Mattson et al. called for a simpler parallelization paradigm: "An ideal solution would automatically exploit concurrency through techniques such as…automatic parallelization of loops." Fortran 2008 [6] answered this call with `do concurrent` and also supported distributed-memory parallelism by incorporating aspects of the Co-Array Fortran language developed in 1996 by Numrich and Reid [7], who stated, "The underlying philosophy of our design is to make the smallest number of changes to the language required to obtain a robust and efficient parallel language without requiring the programmer to learn very many new rules."

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research.

Fortran 2023 greatly expands the parallel feature set. The Cray, Intel, LFortran, LLVM, and NVIDIA compilers automatically parallelize `do concurrent`. The Cray, Intel, GNU, and NAG compilers support coarrays. Thus, language-based parallelism is emerging as a portable alternative to extra-language programming models.

This talk will present experiences with the automatic parallelization of `do concurrent` in the Fortran 2023 deep learning library Inference-Engine[1] and coarray communication in the Intermediate Complexity Atmospheric Research (ICAR) model[2], respectively.

References

[1] T. Mattson and M. Wrinn, "Parallel programming: can we please get it right this time?" in *Proceedings of the 45th annual design automation conference*, 2008, pp. 7–11.

[2] D. J. Becker, T. Sterling, D. Savarese, J. E. Dorband, U. A. Ranawak, and C. V. Packer, "Beowulf: A parallel workstation for scientific computation," in *Proceedings, international conference on parallel processing*, vol. 95, 1995, pp. 11–14.

[3] K. Olukotun, B. A. Nayfeh, L. Hammond, K. Wilson, and K. Chang, "The case for a single-chip multiprocessor," *ACM Sigplan Notices*, vol. 31, no. 9, pp. 2–11, 1996.

[4] J. D. Owens, D. Luebke, N. Govindaraju, M. Harris, J. Krüger, A. E. Lefohn, and T. J. Purcell, "A survey of general-purpose computation on graphics hardware," in *Computer graphics forum*, vol. 26, no. 1. Wiley Online Library, 2007, pp. 80–113.

[5] D. W. Walker and J. J. Dongarra, "Mpi: a standard message passing interface," *Supercomputer*, vol. 12, pp. 56–68, 1996.

[6] Fortran Standards Committee JTC1/SC22/WG5, *Information technology — Programming languages — Fortran, ISO/IEC 1539-1:2010*. International Organization for Standardization (ISO), Oct 2010, https://www.iso.org/standard/50459.html.

[7] R. W. Numrich and J. Reid, "Co-array fortran for parallel programming," in *ACM Sigplan Fortran Forum*, vol. 17, no. 2. ACM New York, NY, USA, 1998, pp. 1–31.

[1] https://go.lbl.gov/inference-engine

[2] https://github.com/berkeleylab/icar