Julia as a unifying end-to-end workflow language on the Frontier exascale system

William F. Godoy, Pedro Valero-Lara, Caira Anderson, Katrina W. Lee, Ana Gainaru,
Rafael Ferreira da Silva, and Jeffrey S. Vetter
Oak Ridge National Laboratory
Oak Ridge, TN, USA
{godoywf,valerolarap,andersonci,leekw,gainarua,silvarf,vetter}@ornl.gov

ABSTRACT

We evaluate the use of Julia as a single language and ecosystem paradigm powered by LLVM for the development of high-performance computing (HPC) workflows components. A Gray-Scott 2-variable diffusion-reaction application using a memory-bound 7-point stencil kernel is run on Frontier, the first exascale supercomputer. We evaluate the feasibility, performance, scaling, and trade-offs of (i) the computational kernel on AMD's MI250x GPUs, (ii) weak scaling up to 4,096 MPI processes/GPUs or 512 nodes, (iii) parallel I/O write using the ADIOS2 library bindings, and (iv) Jupyter Notebooks for interactive data analysis. Our results suggest that although Julia generates a reasonable LLVM-IR kernel, there is nearly a 50% performance difference with native AMD HIP stencil codes on GPU. As expected, we observed near-zero overhead when using MPI and parallel I/O bindings to system-wide installed implementations. Consequently, Julia emerges as a compeling high-performance plus high-productivity workflow composition strategy as measured on the largest supercomputer in the world.

KEYWORDS

Julia, end-to-end workflows, High-Performance Computing, HPC, data analysis, notebooks

ACM Reference Format:
William F. Godoy, Pedro Valero-Lara, Caira Anderson, Katrina W. Lee, Ana Gainaru, Rafael Ferreira da Silva, and Jeffrey S. Vetter. 2023. Julia as a unifying end-to-end workflow language on the Frontier exascale system. In *The International Conference for High Performance Computing, Networking, Storage, and Analysis Workshops (SC-W 2023), November 12–17, 2023, Denver, CO, USA*. ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3605731.3605886

1 INTRODUCTION

The recent emphasis on the end-to-end workflow development process for high-performance computing (HPC) applications acknowledges the increasing complexity required for achieving performance, portability, and productivity [1]. This complexity is primarily driven by two factors: (i) the evolving application requirements for experimental, observational, and computational science; and (ii) the extreme heterogeneity of our computing and data generation and processing systems [3, 4, 7]. Julia provides: (i) a dynamic just-in-time (JIT) compiled front end to LLVM [5], (ii) a lightweight interoperability layer with existing C and Fortran HPC codes, and (iii) a unified community ecosystem (e.g., packaging and testing). The Julia programming language [2] is a valuable alternative in the convergence of high-productivity and high-performance that needs to be tested on exascale hardware. In this work, we measure and analyze the computational performance aspects of a Gray-Scott diffusion-reaction HPC workflow application [6] written in Julia running on Frontier, the first exascale system in the world[1].

2 SIMULATION

Gray-Scott is a two-variable diffusion-reaction three-dimensional model described by the partial differential equation (PDEs) shown in Equations (1a) and (1b).

$$\frac{\partial U}{\partial t} = D_U \nabla^2 U - UV^2 + F\left(1 - U\right) + nr \tag{1a}$$

$$\frac{\partial V}{\partial t} = D_V \nabla^2 V + UV^2 + - \left(F + k\right) V \tag{1b}$$

---

[1] https://www.olcf.ornl.gov/frontier

This manuscript has been authored by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the US Department of Energy (DOE). The publisher acknowledges the US government license to provide public access under the DOE Public Access Plan (https://energy.gov/downloads/doe-public-access-plan).

SC-W 2023, November 12–17, 2023, Denver, CO, USA

ACM ISBN 979-8-4007-0842-8/23/08…$15.00
https://doi.org/10.1145/3605731.3605886

where U and V are the output concentrations of two reacting and diffusing chemicals, while the inputs are listed as follows:

- D_u and D_v are the diffusion rates for U and V
- F is the feed rate of U into the system
- k is the kill rate of V from the system
- n is the magnitude of the noise to be added to the system
- r is a uniformly distributed random number between -1 and 1 for each time and spatial coordinate

As illustrated in Equations (2a) and (2b), the set of governing equations are discretized in time, t, and space, i, j, k, on a regular normalized mesh using a simple forward and central differences, respectively.

$$U^{t+1}_{i,j,k} = U^{t}_{i,j,k} + \Delta t \left[D_U \nabla^2 U^{t}_{i,j,k} + S^{t}_{U} \right] \tag{2a}$$

$$V^{t+1}_{i,j,k} = V^{t}_{i,j,k} + \Delta t \left[D_V \nabla^2 V^{t}_{i,j,k} + S^{t}_{V} \right] \tag{2b}$$

where Δt is an input time step variable, S are the local source terms for U and V defined in Equations (1a) and (1b), and the Laplacian operator ∇^2 is defined in Equation (3) for the 3D "nearest-neighbor" Jacobi 7-point stencil in normalized spatial units:

$$\begin{aligned}\nabla^2 U^{t}_{i,j,k} = -U^{t}_{i,j,k} + \frac{1}{6} \Big[& U^{t}_{i-1,j,k} + U^{t}_{i+1,j,k} + \\ & U^{t}_{i,j-1,k} + U^{t}_{i,j+1,k} + \\ & U^{t}_{i,j,k-1} + U^{t}_{i,j,k+1} \Big].\end{aligned} \tag{3}$$

3 CONCLUSIONS

We present an initial evaluation of the Julia programming language on the Frontier supercomputer, up to 4,096 GPUs and MPI processes, representing 512 nodes. We used a 2-variable diffusion-reaction code, Gray-Scott, to test the performance of the Julia HPC ecosystem in the development of workflow components. As shown in Table 1, the Julia stencil solver achieves close to 50% of the bandwidth of the AMD HIP implementation of a Laplacian kernel on Frontier's MI250x AMD GPU's, hence there is still a need to close performance gaps. Meanwhile, we see in Figure 1, the measured weak scaling due to MPI communication and parallel I/O components suggest that bindings available in Julia are lightweight layers on top of the underlying system MPI and ADIOS-2 library implementations. The Julia implementation shows similar patterns in overhead and variability typical in network and file system communication in HPC systems when measuring weak scalability without I/O as shown in Figure 2. Therefore, the LLVM-based Julia HPC ecosystem presents an attractive alternative for developing co-design components given the high-performance and high-productivity requirements for the end-to-end workflows powering scientific discovery (e.g. AI, FAIR) in exascale systems.

Table 1: Average bandwidth comparison of different stencil implementations on a single GPU.

Kernel	Bandwidth (GB/s)	
	Effective	Total
Julia GrayScott.jl		
- 2-variable (application)	312	570
- 1-variable no random	312	625
HIP single variable	599	1,163
Theoretical peak MI250x	1,600	

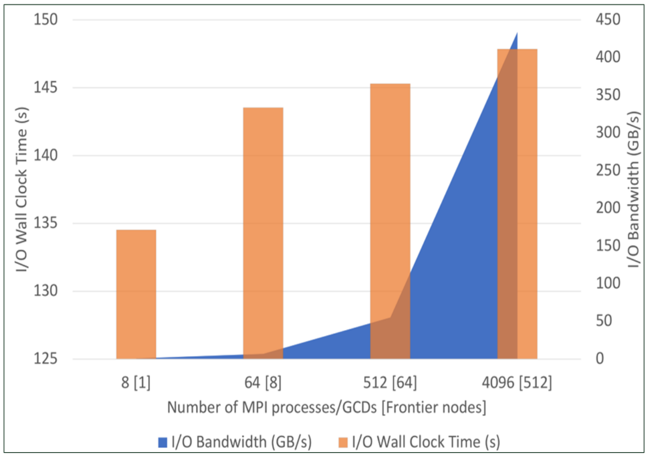

Figure 1: Weak scaling on parallel I/O showing wall-clock times and bandwidths performance using ADIOS2.jl on Frontier

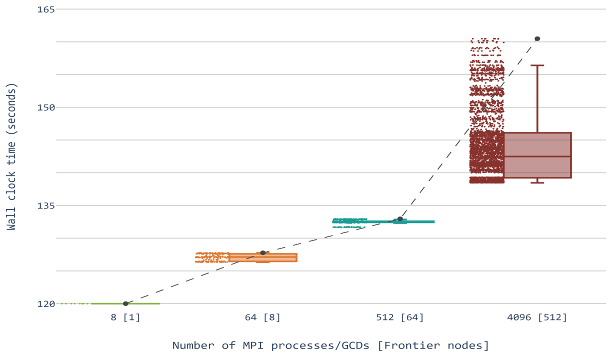

Figure 2: Weak scaling including single MPI process variability obtained with Gray-Scott.jl on Frontier

ACKNOWLEDGMENTS

This research was supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the US Department of Energy Office of Science and the National Nuclear Security Administration. This research used resources of the Oak Ridge Leadership Computing Facility and the Experimental Computing Laboratory (ExCL) at the Oak Ridge National Laboratory, which is supported by the Office of Science of the US Department of Energy under Contract No. DE-AC05-00OR22725.

REFERENCES

[1] Tal Ben-Nun, Todd Gamblin, D. S. Hollman, Hari Krishnan, and Chris J. Newburn. 2020. Workflows are the New Applications: Challenges in Performance, Portability, and Productivity. In *2020 IEEE/ACM International Workshop on Performance, Portability and Productivity in HPC (P3HPC)*. 57–69. https://doi.org/10.1109/P3HPC51967.2020.00011

[2] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B Shah. 2017. Julia: A Fresh Approach to Numerical Computing. *SIAM Rev.* 59, 1 (Jan. 2017), 65–98. https://doi.org/10.1137/141000671

[3] Ewa Deelman, Tom Peterka, Ilkay Altintas, Christopher D Carothers, Kerstin Kleese van Dam, Kenneth Moreland, Manish Parashar, Lavanya Ramakrishnan, Michela Taufer, and Jeffrey Vetter. 2018. The future of scientific workflows. *The International Journal of High Performance Computing Applications* 32, 1 (2018), 159–175. https://doi.org/10.1177/1094342017704893 arXiv:https://doi.org/10.1177/1094342017704893

[4] Rafael Ferreira da Silva, Rosa Filgueira, Ilia Pietri, Ming Jiang, Rizos Sakellariou, and Ewa Deelman. 2017. A Characterization of Workflow Management Systems for Extreme-Scale Applications. *Future Generation Computer Systems* 75 (2017), 228–238. https://doi.org/10.1016/j.future.2017.02.026

[5] Chris Lattner and Vikram Adve. 2004. LLVM: A compilation framework for lifelong program analysis & transformation. In *International Symposium on Code Generation and Optimization, 2004. CGO 2004.* IEEE, 75–86.

[6] John E. Pearson. 1993. Complex Patterns in a Simple System. *Science* 261, 5118 (1993), 189–192. https://doi.org/10.1126/science.261.5118.189 arXiv:https://www.science.org/doi/pdf/10.1126/science.261.5118.189

[7] Jeffrey S. Vetter, Ron Brightwell, Maya Gokhale, Pat McCormick, Rob Ross, John Shalf, Katie Antypas, David Donofrio, Travis Humble, Catherine Schuman, Brian Van Essen, Shinjae Yoo, Alex Aiken, David Bernholdt, Suren Byna, Kirk Cameron, Frank Cappello, Barbara Chapman, Andrew Chien, Mary Hall, Rebecca Hartman-Baker, Zhiling Lan, Michael Lang, John Leidel, Sherry Li, Robert Lucas, John Mellor-Crummey, Paul Peltz Jr., Thomas Peterka, Michelle Strout, and Jeremiah Wilke. 2018. Extreme Heterogeneity 2018 - Productive Computational Science in the Era of Extreme Heterogeneity: Report for DOE ASCR Workshop on Extreme Heterogeneity. (12 2018). https://doi.org/10.2172/1473756