OAK RIDGE National Laboratory

June 19, 2025

Exploring Performance Portability in Julia and Mojo

Tatiana Melnichenko

University of Tennessee, Knoxville

U.S. DEPARTMENT of ENERGY

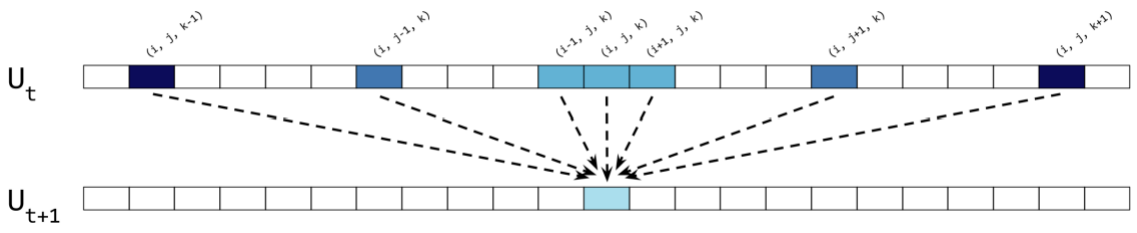
ORNL IS MANAGED BY UT-BATTELLE LLC
FOR THE US DEPARTMENT OF ENERGY

Introduction: Tatiana Melnichenko

- Senior Undergrad at UTK in Computer Science
- Research Assistant at Innovative Computing Laboratory
- ORNL SULI intern, mentored by Dr. William Godoy
- Focus: productivity and performance of high-level languages, like Julia and Mojo
- **Goal: Make HPC more accessible to people with little or no programming experience**

From Zero Julia to JACC Kernels in 3 Weeks

- Background: C/C++, OpenMP, CUDA, HIP
- Started with zero Julia experience
- Learned via the same tutorial notebooks you're using
- Developed a 7-point-stencil in JACC (Julia for Accelerators: https://ieeexplore.ieee.org/document/10820713)
- Also implemented the same kernel in Mojo



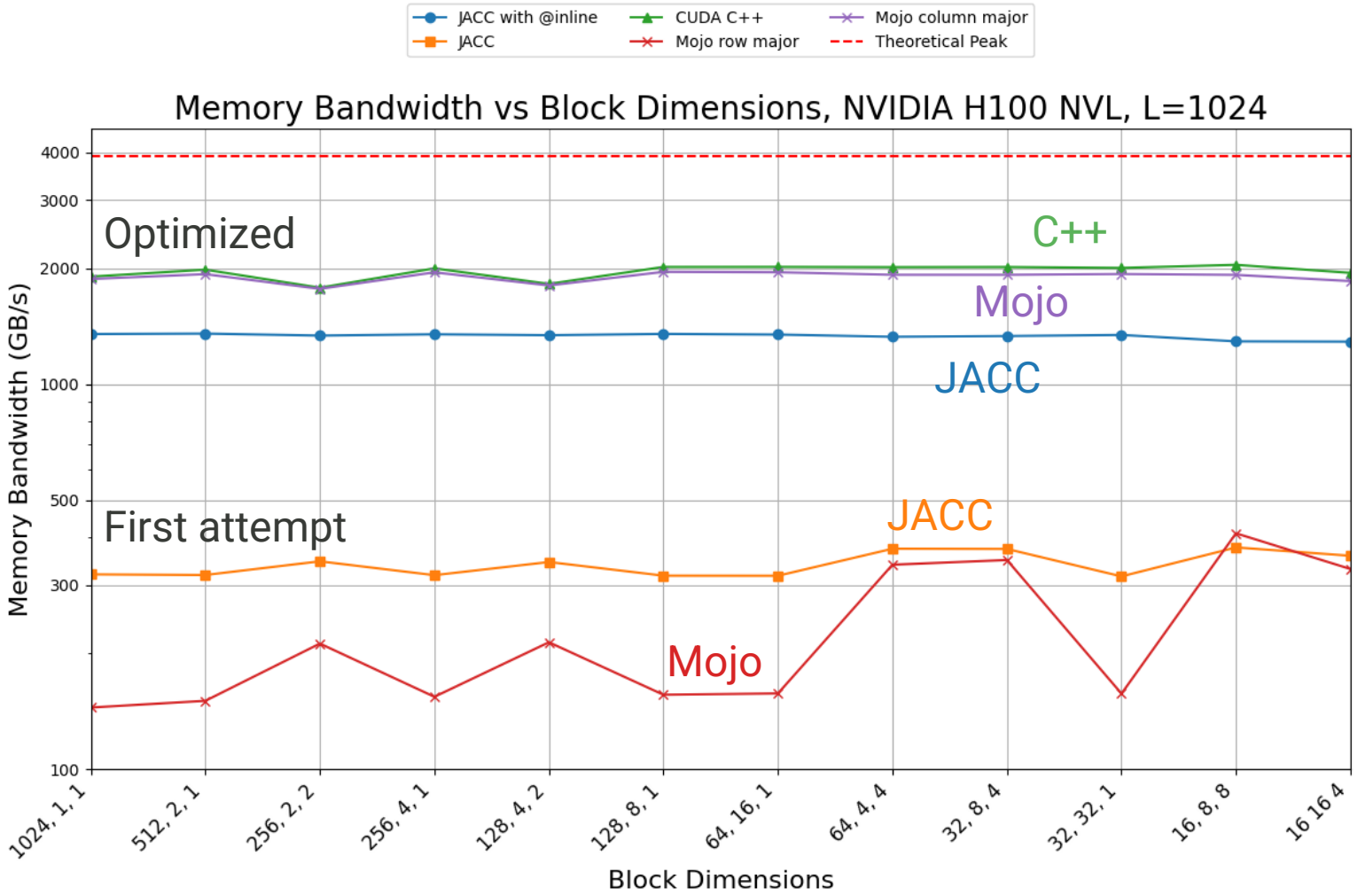

```julia
module Laplacian
import JACC
JACC.@init_backend

function init_test_kernel!(i, j, k, u, nx, ny, nz, hx, hy, hz)
    if i <= nx && j <= ny && k <= nz
        c = 0.5
        x = (i - 1) * hx
        y = (j - 1) * hy
        z = (k - 1) * hz
        Lx = nx * hx
        Ly = ny * hy
        Lz = nz * hz
        @inbounds u[i, j, k] = c * x * (x - Lx) + c * y * (y - Ly) + c * z * (z - Lz)
    end
end

@inline function calculate_kernel!(i, j, k, u, f, nx, ny, nz,
                                   invhx2, invhy2, invhz2, invhxyz2)
    if i > 1 && i < nx &&
       j > 1 && j < ny &&
       k > 1 && k < nz
        @inbounds f[i, j, k] = u[i, j, k] * invhxyz2
                             + (u[i - 1, j    , k    ] + u[i + 1, j    , k    ]) * invhx2
                             + (u[i    , j - 1, k    ] + u[i    , j + 1, k    ]) * invhy2
                             + (u[i    , j    , k - 1] + u[i    , j    , k + 1]) * invhz2
    end
end
```

Performance Comparison: Julia vs Mojo

- Benchmarked stencil JACC, Mojo, and C++ CUDA kernels
- 1024 threads organized in different 1D/2D/3D block layouts
- Performance metric: memory bandwidth (GB/s)
- Future goal: Auto-tuning block layouts in JACC



Memory Bandwidth: Higher is Better!

Thank you for attention!

Acknowledgements:

- This work was supported in part by the U.S. Department of Energy, Office of Science, Office of Workforce Development for Teachers and Scientists (WDTS) under the Science Undergraduate Laboratory Internships Program (SULI).
- This research used resources of the Oak Ridge Leadership Computing Facility and the Experimental Computing Laboratory 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.