
OAK RIDGE National Laboratory

JACC
JULIA FOR ACCELERATORS

# Leveraging HPC Meta-Programming and Performance Portability with the Just-in-Time and LLVM-based Julia Language

PhD. Pedro Valero-Lara

Senior Computer Scientist, valerolara@ornl.gov

U.S. DEPARTMENT OF ENERGY

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

OAK RIDGE National Laboratory

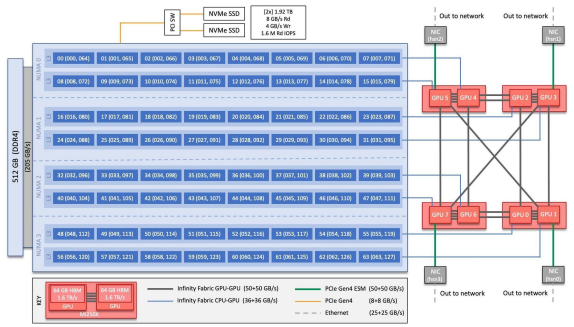
Motivation:

# Performance Portability and Programming Productivity

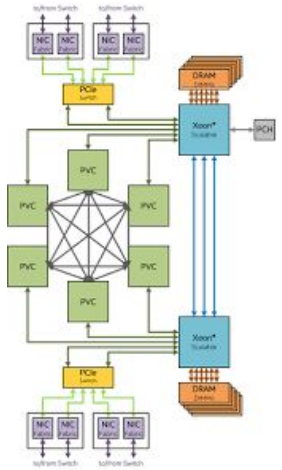
## Program once and deploy everywhere !



FRONTIER



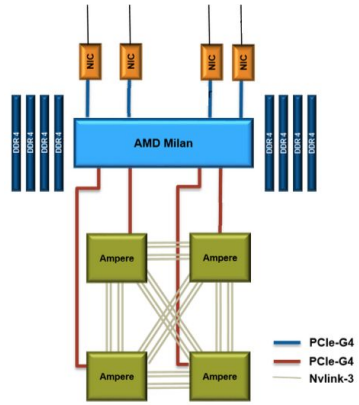
AMD ROCm



Aurora

oneAPI

SYCL™


NVIDIA CUDA

OpenMP

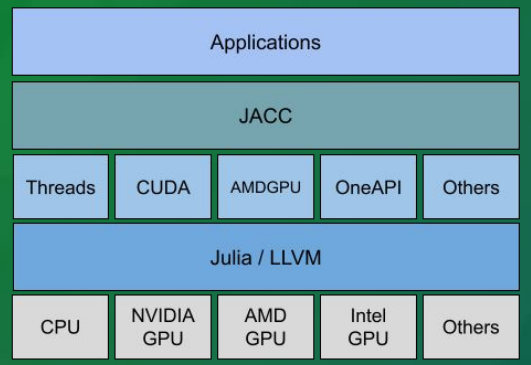
OAK RIDGE National Laboratory

JACC
JULIA FOR ACCELERATORS

Julia for ACCelerators (JACC), What is that??

# *Think in Kokkos, but now imagine that it is easy to use*

**https://github.com/JuliaORNL/JACC.jl**




U.S. DEPARTMENT OF ENERGY

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

# JACC Model, How to use it??

***Descriptive, not prescriptive!***

JACC.Array

- An alias to the corresponding backend memory management

JACC.parallel_for and JACC.parallel_reduce

- Kernel passed an argument
- Uni-dimensional and multi-dimensional APIs

JACC.ones or JACC.zeros

JACC.shared

- Exploit on-chip GPUs shared memory

JACC.multi

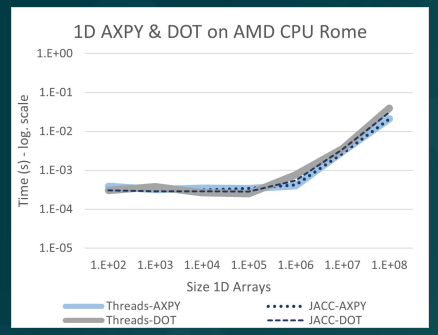
- Support for multi-GPU computing

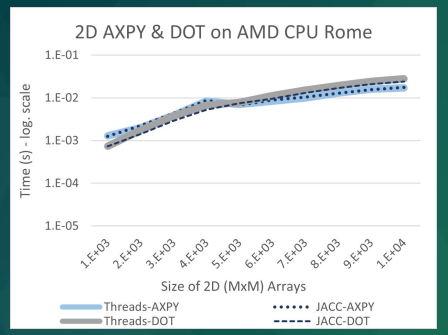
```
# Unidimensional arrays
function axpy(i, alpha, x, y)
  x[i] += alpha * y[i]
end
function dot(i, x, y)
  return x[i] * y[i]
end
SIZE = 1_000_000
x = round.(rand(Float64, SIZE) * 100)
y = round.(rand(Float64, SIZE) * 100)
alpha = 2.5
dx = JACC.Array(x)
dy = JACC.Array(y)
JACC.parallel_for(SIZE, axpy, alpha, dx, dy)
res = JACC.parallel_reduce(SIZE, dot, dx, dy)
# Multidimensional arrays
function axpy(i, j, alpha, x, y)
  x[i,j] = x[i,j] + alpha * y[i,j]
end
function dot(i, j, x, y)
  return x[i,j] * y[i,j]
end
SIZE = 1_000
x = round.(rand(Float64, SIZE, SIZE) * 100)
y = round.(rand(Float64, SIZE, SIZE) * 100)
alpha = 2.5
dx = JACC.Array(x)
dy = JACC.Array(y)
JACC.parallel_for((SIZE,SIZE),axpy,alpha,dx,dy)
res = JACC.parallel_reduce((SIZE,SIZE),dot,dx,dy)
```

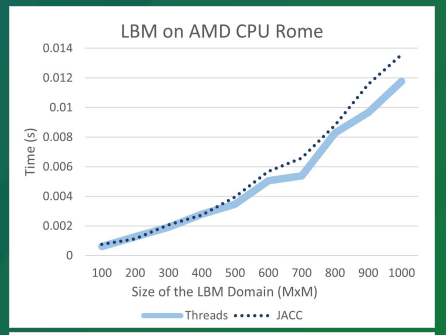
# OK! But this is HPC, What about performance??

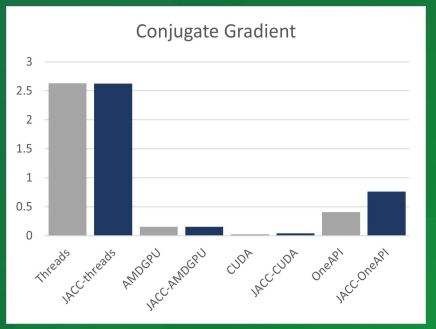
JACC
JULIA FOR ACCELERATORS

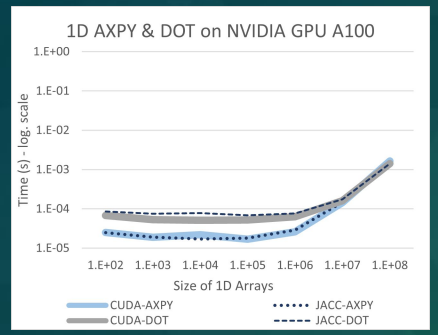


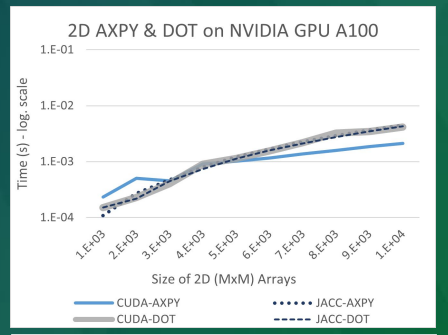



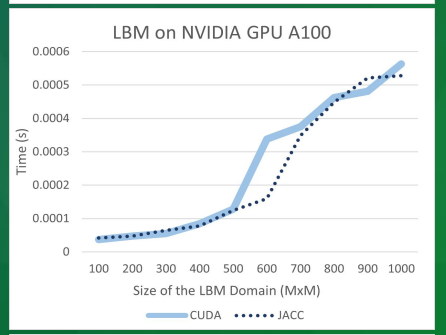



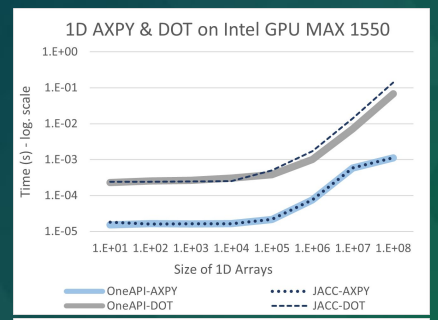



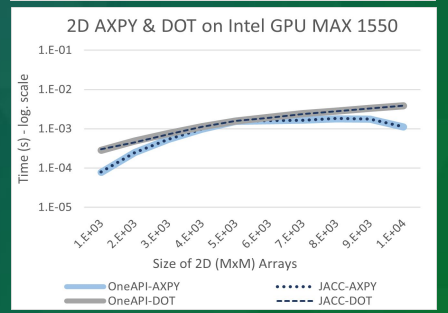



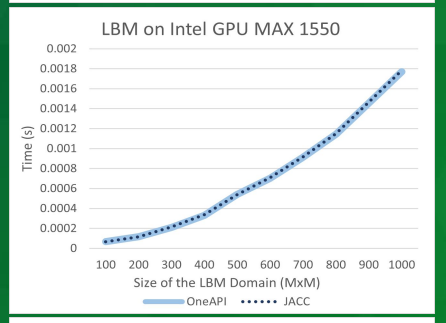



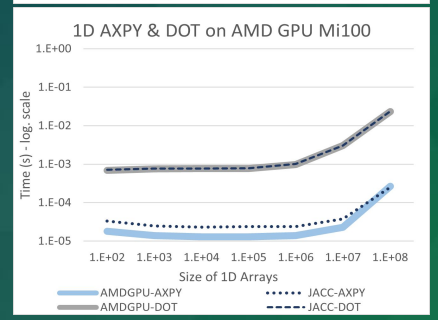



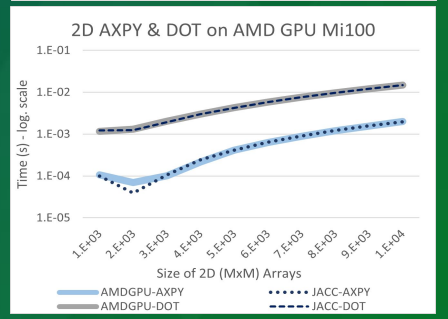



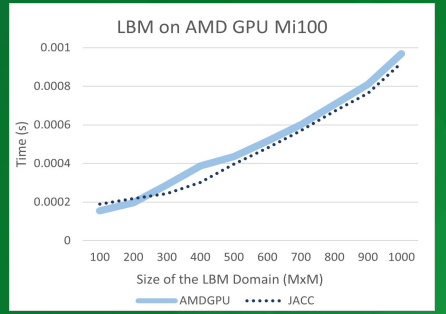



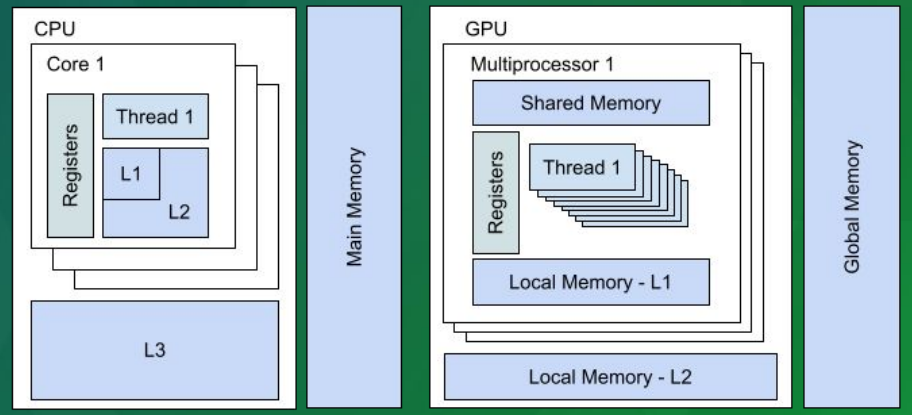
# JACC.shared: Exploiting High-Bandwidth on-chip GPUs Memory

***Shared memory on-chip GPUs memory are orders of magnitude faster than global memory***

***Shared memory is useful to accelerate computations where multiple threads (in the same block) must access the same data multiple times***

JACC.shared: Y = JACC.shared(X)

- X is a JACC.Array and Y is a copy of X array stored in on-chip shared memory GPU
- X can be an array of any dimension
- Y can only be a unidimensional array
- To be used inside functions
- The array passed as argument must fit the capacity of the shared memory




```
function spectral(i, j, image, filter, num_bands)
 for b in 1:bands
    @inbounds image[b, i, j] *= filter[j]
  end
end

function spectral_shared(i, j, image, filter,
    num_bands)
 #Shared memory initialization
 filter_shared = JACC.shared(filter)
 for b in 1:bands
  @inbounds image[b, i, j] *= filter_shared[j]
 end
end

num_bands = 60
num_voxel = 10_240
size_voxel = 64*64
image = init_image(Float32,
        num_bands, num_voxel, size_voxel)
filter = init_filter(Float32, size_voxel)
jimage = JACC.Array(image)
jimage_shared = JACC.Array(image)
jfilter = JACC.Array(filter)
JACC.parallel_for((num_voxel,size_voxel),
     spectral[_shared], jimage, jfilter, num_bands)
```

OAK RIDGE National Laboratory

Julia for ACCelerators (JACC)

# *Conclusions*
# *What's next??*

**https://github.com/JuliaORNL/JACC.jl**

JACC
JULIA FOR ACCELERATORS

U.S. DEPARTMENT OF ENERGY

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

OAK RIDGE National Laboratory

JACC
JULIA FOR ACCELERATORS

# Thanks!!
# Questions??


PhD. Pedro Valero-Lara

Senior Computer Scientist, valerolara@ornl.gov


U.S. DEPARTMENT OF ENERGY

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