AMDGPU.jl

By Julian Samaroo

State of Affairs

- Deploying core ROCm artifacts as JLLs
- Passing GPUArrays testsuite
- Refactor all the things
- Lots of new AMD supercomputers coming online
  - Lots of new users!
  - Lots of bugs found…

ORNL Frontier

CSCS LUMI

Pawsey Setonix

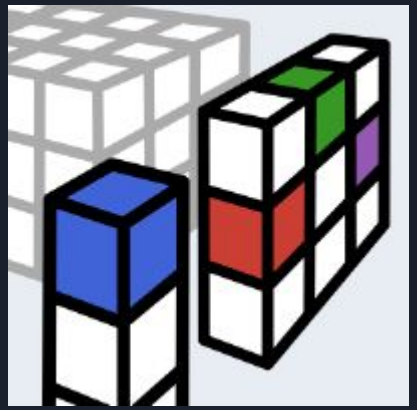
ROCm Artifacts as JLLs

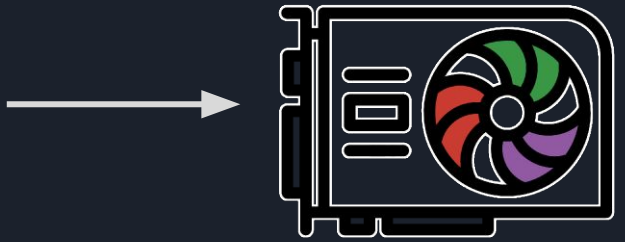
- Source builds suck, and ROCm is not available in all distros
- We have working ROCm builds as JLLs, and are shipping them via AMDGPU.jl
- Not all packages are available as JLLs, but we have the core libraries setup
- ROCm libraries:
  - rocRAND - JLL available
  - rocBLAS - Almost ready
  - rocFFT - Almost ready
  - rocSPARSE - Not started, needs rocBLAS
  - rocSOLVER - Not started
  - MIOpen - Not started
- Various other libraries/tools could also be shipped as JLLs

Yggdrasil

Test Coverage

- The ROCArray needs to be generic enough to work with ecosystem libraries
- We need a conformance test - GPUArrays.jl has a testsuite which is very comprehensive
- Thanks to Valentin Churavy, we now pass the entire testsuite
- What this gets us:
  - Automatic composability with many generic libraries
  - Fast broadcast implementation
  - Drop-in replacement wherever CuArrays are used

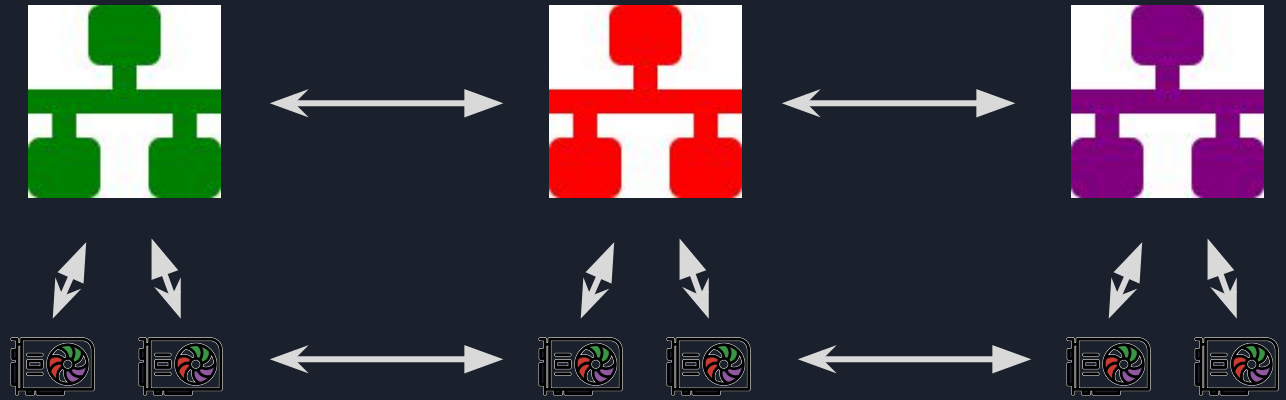




MPI Support

- MPI backs a large number of scientific applications, and runs well on supercomputers
- Many Julia users use MPI through MPI.jl, DistributedArrays.jl, and ImplicitGlobalGrid.jl
- CUDA-aware MPI is a thing, so why not ROCm-aware MPI?
- This is now a thing! (Thanks to Ludovic Räss and Simon Byrne)
- This will enable greater usage of supercomputer resources and faster code

ROCm-aware MPI



Future Developments

- Examples - We need more of these!
- Library Integrations
  - More ROCm JLLs
  - Integrate with Julia ecosystem packages (SciML, Flux/Lux, etc.)
- Profiling and Debugging support
  - ROCm has open-source profile/debugger libraries that we can use
  - GDB sucks, can we do better?
- Performance enhancements
- Fancy features
  - Device-side kernel launch
  - Device-to-device IPC

See you soon!

- AMDGPU.jl is ready to tackle your computational problems
- We have a strong base of users helping squash bugs and optimize performance
- AMD engineers have been helping out with development and ROCm JLLs
- We’re ready to run at supercomputer scale!