

# The Frontier Programming Environment at OLCF

David E. Bernholdt

Programming Environment and Tools Lead  
Oak Ridge Leadership Computing Facility

Oak Ridge National Laboratory



ORNL is managed by UT-Battelle LLC for the US Department of Energy

# Overview



# Contributors to Frontier Programming Environment

## Vendor-Provided

- Cray Programming Environment (CPE)
  - Includes Cray compiler for C, C++, and Fortran plus GCC compiler. All the Cray profiling, tuning, and debugging tools. OpenMP and Cray MPI optimized for AMD GPU direct.
- AMD ROCm programming environment
  - Includes LLVM compiler to generate optimized code for both the AMD Trento CPU and MI200 GPU. It will support: C, C++, and Fortran and have GPU offload support. HIP, a CUDA-like direct GPU programming model (with CUDA to HIP conversion utilities).

## Other Sources

- ECP
  - LLVM enhancements: Flang (Fortran front-end), OpenMP, OpenACC
  - Kokkos and RAJA
  - HIP LZ (HIP support for Aurora)
  - MPI, HPCToolkit, PAPI enhancements
  - ...
- ALCF + OLCF
  - Pilot implementation of DPC++/SYCL for Frontier
- OLCF
  - GCC enhancements to better support OpenACC, OpenMP, Fortran on Summit and Frontier

# Programming Environment

- Compilers Offered

- Cray PE (C/C++ LLVM-based; Cray Fortran)
- AMD ROCm (LLVM-based)
- GCC

Items in green are also available on Summit

- Programming Languages & Models Supported (in which compilers)

- C, C++, Fortran (all)
- OpenACC (GCC) 2.6 substantially complete, 2.7 planned
- OpenMP (all) 5.0-5.2 in progress – most priority features complete, details vary
- HIP (Cray, AMD)
- Kokkos/RAJA (all)
- UPC (Cray, GCC)

- Transition Paths

- CUDA: semi-automatic translation to HIP
- CUDA Fortran: HIP kernels called from Fortran (a more portable approach)
  - CUDA Fortran kernels need to be translated to C++/HIP (manual process)
  - Fortran bindings to HIP and ROCm libraries and HIP runtime available through AMD's hipfort project

# Programming Tools

## Debuggers and Correctness Tools

Tool
<i>System-Level Tools</i>
Arm DDT
Cray CCDB
Cray ATP
STAT
<i>Node-Level Tools</i>
ROCgdb
Cray GDB4HPC

Items in green are also available on Summit

## Performance Tools

Tool
<i>System-Level Tools</i>
Arm MAP/Performance Reports
CrayPat/Apprentice2 (Cray)
Reveal (Cray)
TAU
HPCToolkit
Score-P / VAMPIR
<i>Node-Level Tools</i>
gprof
PAPI
ROCprof
ROC-profiler & ROC-tracer libraries

# Scientific Libraries and Tools

Functionality	CPU	GPU	Notes
<b>BLAS</b>	<b>Cray LibSci, AMD BLIS, PLASMA</b>	<b>Cray LibSci_ACC, AMD roc/hipBLAS, AMD rocAMD ROCm Tensile, MAGMA</b>	MAGMA and PLASMA are open source software led by the UTK Innovative Computing Laboratory
<b>LAPACK</b>	<b>Cray LibSci, AMD libFlame, PLASMA</b>	<b>Cray LibSci_ACC, AMD roc/hipSolver, MAGMA</b>	
<b>ScaLAPACK</b>	Cray LibSci	ECP SLATE, Cray LibSci_ACC	
<b>Sparse</b>		<b>AMD roc/hipSparse, AMD rocALUTION</b>	
Mixed-precision iterative refinement	Cray IRT, MAGMA	MAGMA	
<b>FFTW</b> or similar	<b>Cray, AMD, ECP FFTX, FFT-ECP</b>	<b>AMD rocFFT, ECP FFTX, FFT-ECP</b>	FFT-ECP focuses on 3D FFTs
<b>PETSc, Trilinos, HYPRE, SUNDIALS, SuperLU</b>			Spack recipes from ECP xSDK

Functionality in **green** is also available on Summit

# Digging a Little Deeper



# For C/C++ Codes

- Multiple compilers available
  - AMD
  - Cray
  - LLVM
- But they're all based on LLVM
  - HPE and AMD are among the many organizations contributing to the development of LLVM
  - Most work is “upstreamed” (contributed to the core LLVM source)
    - But not everything is accepted (immediately), or may be held back as proprietary
  - Capabilities (and bugs) are likely to be generally similar at any point in time...
  - But not identical!
- LLVM is also available on Summit



# For Fortran Codes

- One useful compiler available at present
  - Cray
    - *Not* based on LLVM
- AMD provides a Fortran implementation, but we don't recommend it
  - It is based on “classic Flang”, in the LLVM ecosystem
  - Support for both the latest language standards and OpenMP offload are limited
- There is extensive work underway in the LLVM community on Flang, but it will be some time before it is production quality

# But What About GCC?

- On this slide “gcc” refers to the whole suite, including gfortran
- OLCF is working with Siemens to implement OpenMP in gcc
- OLCF will provide recent release and development versions of gcc on Frontier
- For various reasons, you should *not* expect gcc-generated executables to be performant for offload at this time
  - Results will vary
  - We are interested in improving the performance of gcc. If you have a troublesome case, reach out to me. (No guarantees, however)
- **GCC is also available on Summit**

# For HIP (and CUDA) Codes

- HIP runs today on AMD and NVIDIA GPUs
- An ECP project is working on supporting HIP on Intel GPUs too
- Recommend a one-time translation of CUDA codes to HIP and make the HIP version primary from then on
- Both Cray and AMD compilers support HIP
  - They both use the AMD runtime
- More on HIP available in the OLCF [Training Archive](#)
- HIP is also available on Summit

# For OpenMP Codes

- OpenMP is very much a work in progress in the LLVM community
  - Most of 5.0 is implemented
  - Parts of 5.1, 5.2 are implemented
- We (DOE labs, including ORNL/OLCF) are trying to help prioritize the order of implementation based on what users tell us they need/want
  - So if you could really use features that aren't available yet, please let us know!
- Cray and AMD compilers use different OpenMP runtimes
- Remember that Cray Fortran is not based on LLVM
- OpenMP implementation in gcc is also a work in progress
- More on OpenMP available in the [OLCF Training Archive](#)
- OpenMP is available on Summit, but different progress on impl

# For OpenACC Codes

- OLCF provides OpenACC support via gcc
  - 2.6 currently supported
  - 2.7 planned
  - 3.x not currently planned – let us know if there are particular features that you could really use
  - Don't expect this to be performant at present
- Cray Fortran supports OpenACC 2.0
  - Work is underway to 3.2 (latest) but no timeline has been given
  - No OpenACC in Cray C/C++
- Work is also underway in the LLVM community on OpenACC
  - Unknown when these will be production
- OpenACC is also available on Summit

# What about SYCL?

- OLCF and ALCF have partnered with Codeplay on a pilot implementation of the Intel DPC++ compiler for AMD GPUs
  - ALCF has also partnered with NERSC on NVIDIA support
- Pilot implementation is complete
  - ~“50%” level of support
  - Tested with a small set of benchmarks and mini-apps
  - Should be available any day now on Spock, with Crusher/Frontier soon to follow
- Seeking interested users to try out the pilot implementation
  - Provide feedback
  - Shake out issues
  - Provide motivation to complete the port
- Can make SYCL available on Summit if there is interest

# Help Us Help You...

- If you have a liaison, work with them
- **If you encounter an issue, file a ticket with OLCF** – otherwise the facility won't (necessarily) know about it, and can't track it
  - Summit, Spock, Crusher, Frontier...
- Take advantage of training events
  - *Preparing for Frontier* series in the OLCF [Training Archive](#)
  - If you have Crusher access: office hours, hackathons, additional trainings