



AMD Presentation Roadmap

HLRS Training Workshop
Bob Robey
Data Center GPUs, AMD

AMD 
together we advance_

Training Workshop Guidelines

– help us create a positive environment

Please help us create a positive collaborative environment for this event. Our top goal is to attract and build the community in high performance computing. Please consider not only the technical aspects of your interactions, but also how it affects other participants.

- Start your comments with a positive statement before your comment or question
- Give equal time to all participants
- Please, help others with technical issues and without negative comments
- And most importantly, respect all participants, regardless of background, language, culture

Day 1 Schedule – Monday, 5 May 2025

Time (UTC+1)	Topic	Presenter
13:00	Host Organization Intro	HLRS
13:10	AMD Presentation Roadmap and Intro to System for Exercises	Bob Robey
13:20	Programming Model for MI200 and MI300 Series	Bob Robey
13:45	Programming Model Exercises	Bob Robey
14:00	----- Coffee Break -----	---
14:10	Introduction to OpenMP® Offloading with USM on the APU	Johanna Potyka
14:55	----- Break -----	---
15:10	OpenMP® Exercises	Johanna Potyka
15:40	Introduction OpenMP Offloading on discrete GPUs	Giacomo Rossi
16:10	OpenMP Exercises Part II	Giacomo Rossi
16:50	Wrapup	Bob Robey

Day 2 Schedule – Tuesday, 6 May 2025

Time (UTC+1)	Topic	Presenter
13:00	HIP and ROCm	Giacomo Capodaglio
14:00	HIP Exercises	Giacomo Capodaglio
14:15	----- Break -----	
14:30	Porting code to HIP	Giacomo Capodaglio
14:50	Porting Exercises	Giacomo Capodaglio
15:00	OpenMP and HIP Interoperability	Johanna Potyka
15:25	Interoperability Exercises	Johanna Potyka
15:40	----- Break -----	
15:55	Optimizing HIP Code	Gina Sitaraman
16:35	HIP Optimization Exercises	Gina Sitaraman
16:55	Wrap-up	Bob Robey

Day 3 Schedule – Wednesday, 7 May 2025

Time (UTC+1)	Topic	Presenter
13:00	Performance Portability Frameworks: Introduction to Kokkos	Bob Robey
13:30	Kokkos Exercises	Bob Robey
13:50	----- Break -----	---
14:00	GPU-Aware MPI on AMD GPUs	Bob Robey
14:30	MPI Exercises	Bob Robey
14:50	----- Break -----	---
15:00	C++ Standard Parallelism	Alessandro Fanfarillo
15:30	C++ Standard Parallelism Exercises	Alessandro Fanfarillo
15:50	Break	
16:00	ML/AI on AMD GPUs	Sidafa Conde
16:30	ML/AI Exercises	Sidafa Conde
16:55	Wrapup	Bob Robey

Day 4 Schedule – Thursday, 8 May 2025

Time (UTC+1)	Topic	Presenter
13:00	Debugging with Rocgdb	Giacomo Capodaglio
13:40	Rocgdb Exercises	Giacomo Capodaglio
14:00	----- Break -----	
14:15	GPU Profiling – Performance Timelines	Luka Stanisic
14:55	Timeline Profiling Exercises	Luka Stanisic
15:15	----- Break -----	---
15:30	Kernel Profiling	Luka Stanisic
16:15	Kernel Exercises	Luka Stanisic
16:45	Additional Training Resources	Bob Robey
16:55	Wrap-up	Bob Robey

Notes

- Preferred Contact: Reach out on slack during and immediately after this course
- Johanna Potyka, Johanna.Potyka@amd.com
- Bob Robey, Bob.Robey@amd.com
- Giacomo Rossi, Giacomo.Rossi@amd.com
- Giacomo Capodaglio, Giacomo.Capodaglio@amd.com
- Gina Sitaraman, Gina.Sitaraman@amd.com
- Sidafa Conde, Sidafa.Conde@amd.com
- Luka Stanisic, Luka.Stanisic@amd.com
- Alessandro Fanfarillo, Alessandro.Fanfarillo@amd.com
- Hands-on Exercises
 - Log in to the AAC node then: git clone <https://github.com/AMD/HPCTrainingExamples>

Introduction to AMD Accelerator Cloud (AAC) for Exercises

Contacts for System Issues -- AAC

- Use the slack channel #cluster for getting help with aac access
- Please contact Bob Robey Bob.Robey@amd.com and Giacomo Capodaglio Giacomo.Capodaglio@amd.com about AAC access requests with the information requested in the excel template sent to you as soon as possible, if not done already.

AAC notes

- You should have gotten an invite to participate in the workshop
 - Problems, request support by email to Bob.Robey@amd.com, Giacomo.Capodaglio@amd.com or in the Slack channel #cluster
- AMD Accelerator Cloud (AAC) Training Container
 - Ubuntu® 22.04
 - ROCm version 6.4.0
 - GCC is version 11
 - Clang/LLVM is version 19

File spaces

- Home Directory
 - Persistent storage is at `/home/aac/shared/teams/dcgpu_training/hlrs/$USER`. Your home directory will be set to this directory.
 - `$HOME=/home/aac/shared/teams/dcgpu_training/hlrs/$USER`
 - Files in your home directory should persist across container restarts and be available from another container with the same userid on systems at the same hosting location.

Logging into AAC

- We use ssh public keys for accessing the AAC system
- First set up a key on your system
- `ssh-keygen -t ed25119`
 - Hit return to take the defaults
- Copy the key to the training node
 - `ssh-copy-id -i ~/.ssh/id_ed25119.pub -p 7000 -o UpdateHostKeys=yes <username>@aac6.amd.com`
- You will get a prompt to enter your password to copy the key into your home directory
- After that you should be able to log onto the system without a password
- You can do this from the terminal in Windows, from wsl, from Linux or MacOS

Logging into AAC

- Accessing AAC
- `ssh <USERNAME>@aac6.amd.com -p <port_number> -i <identityfile>`
- To simplify the login to AAC, you can add the following to your `.ssh/config` file.
- # AAC
 - Host aac
 - User <USERNAME>
 - Hostname aac1.amd.com
 - #IdentityFile <path>/<private key file> # customize location of key file
 - IdentityFile id_ed25519 # file is in .ssh directory
 - ServerAliveInterval 600
 - ServerAliveCountMax 30
- The `ServerAlive*` lines in the config file may be added to avoid timeouts when idle.
- Now you can login with `ssh aac -p <port_number>`

File copying

You should be able to copy files in or out with the `scp` command.

- In: `scp -i <path>/<keyfile> -P <port_number> <file> USER@aac6.amd.com:~/<path>/<file>`
- Out: `scp -i <path>/<keyfile> -P <port_number> USER@aac6.amd.com:~/path/to/your/file ./`

You can also use `rsync` command

- `rsync -avz -e "ssh -i <path>/<keyfile> -p <port_number>" <file> <USER>@aac6.amd.com:~/path/to/your/files`

Environment -- modules

Modules using Lua modules

- **Linux[®]**
 - clang/base -- Clang/LLVM standard compiler installations
 - gcc/base -- GCC standard compiler installations with offload modules
 - miniconda3/24.9.2 – Python[™] environment
 - miniforge3/24.9.0 – alternate Python[™] environment
- **ROCm**
 - rocm/6.4.0 -- ROCm software stack including hip and hip libraries
 - amdclang/19.0.0-6.4.0 -- OpenMP[®] compiler
 - amdflang-new/rocm-afar-6.0.0 -- next gen Fortran compiler
 - hipfort/6.4.0-- Fortran wrappers for hip calls
 - opencl/6.4.0 – OpenCL[™] language
 - rocprofiler-compute/6.4.0 Kernel profiler, built-in version
 - rocprofiler-systems/6.4.0 Timeline profiler, built-in version
- **ROCmPlus -- MPI**
 - openmpi/5.0.7-ucc1.3.0-ucx1.18.0 – OpenMPI with GPU-aware MPI
 - mpi4py/4.0.3 – python interface to MPI

Environment -- modules

Modules using Lua modules

- ROCmPlus -- Latest Compilers
 - hipfort_from_source/6.4.0
- ROCmPlus-AI
 - cupy/14.0.0a1
 - jax/0.4.35
 - pytorch/2.7.0
- Misc
 - fftw/3.3.10
 - hdf5/1.14.5
 - hipifly/dev
 - hpctoolkit/2024.01.99-next
 - hypre/2.33.0
 - kokkos/4.6.00
 - netcdf-c/4.9.3-rc1
 - netcdf-fortran/4.6.2-rc1
 - petsc/3.23.0
 - scorep/9.0
 - tau/dev

Environment -- modules

Modules using Lua modules

- ROCmPlus -- Latest Compilers
 - [amdflang-new](#)

AMD is working on a Fortran compiler and soon will include it in ROCm. During the training, **HLRS will have hands-on access to the beta**. The beta itself contains early access to open-sourced pre-release features. Going forward, AMD is able to provide continued early access via AFARs (Advanced Feature Access Releases) as improvements and features are added to the beta. As with any beta, we looking to gather feedback on functionality, usability and user experience.

PRE-PRODUCTION SOFTWARE: The software accessible on this page may be a pre-production version, intended to provide advance access to features that may or may not eventually be included into production version of the software. Accordingly, pre-production software may not be fully functional, may contain errors, and may have reduced or different security, privacy, accessibility, availability, and reliability standards relative to production versions of the software. Use of pre-production software may result in unexpected results, loss of data, project delays or other unpredictable damage or loss. Pre-production software is not intended for use in production, and your use of pre-production software is at your own risk.

Environment -- modules

Modules using Lua modules

Compiler modules set the CC, CXX, FC flags. Only one compiler module can be loaded at a time. hipcc is in the path when the rocm module is loaded. These modules also set the OMPI_CC/CXX/FC environment variables so that the MPI compiler wrappers use the currently set compiler.

Module commands

- module avail
- module load <package>
- module unload <package>

Environment -- Slurm

Slurm configuration with single queue LocalQ

sinfo

PARTITION	AVAIL	TIMELIMIT	NODES	STATE	NODELIST
LocalQ	up	2:00:00	1	idle	localhost

Slurm commands

- salloc can be used to schedule a long-term interactive session
- sbatch is used to submit a job to the batch queue
- squeue will show the jobs running in the batch queue
- The slurm configuration is sort of a combination of a login node and backend compute resources
 - Queue resources are the same as the local node
 - GPUs are still available without going through Slurm

Installed software

- emacs
- vim
- autotools
- cmake
- tmux
- boost
- eigen
- fftw
- gmp
- gsl
- hdf5-openmpi
- lapack
- magma
- matplotlib
- parmetis
- mpfr
- mpi4py
- openblas
- openssl
- swig
- numpy
- scipy
- h5sparse

Disclaimer

The information presented in this document is for informational purposes only and may contain technical inaccuracies, omissions, and typographical errors. The information contained herein is subject to change and may be rendered inaccurate for many reasons, including but not limited to product and roadmap changes, component and motherboard version changes, new model and/or product releases, product differences between differing manufacturers, software changes, BIOS flashes, firmware upgrades, or the like. Any computer system has risks of security vulnerabilities that cannot be completely prevented or mitigated. AMD assumes no obligation to update or otherwise correct or revise this information. However, AMD reserves the right to revise this information and to make changes from time to time to the content hereof without obligation of AMD to notify any person of such revisions or changes.

THIS INFORMATION IS PROVIDED 'AS IS.' AMD MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND ASSUMES NO RESPONSIBILITY FOR ANY INACCURACIES, ERRORS, OR OMISSIONS THAT MAY APPEAR IN THIS INFORMATION. AMD SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL AMD BE LIABLE TO ANY PERSON FOR ANY RELIANCE, DIRECT, INDIRECT, SPECIAL, OR OTHER CONSEQUENTIAL DAMAGES ARISING FROM THE USE OF ANY INFORMATION CONTAINED HEREIN, EVEN IF AMD IS EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

© 2025 Advanced Micro Devices, Inc. All rights reserved. AMD, the AMD Arrow logo, ROCm, Radeon and combinations thereof are trademarks of Advanced Micro Devices, Inc. Other product names used in this publication are for identification purposes only and may be trademarks of their respective companies.

Git and the Git logo are either registered trademarks or trademarks of Software Freedom Conservancy, Inc., corporate home of the Git Project, in the United States and/or other countries

Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

The OpenMP name and the OpenMP logo are registered trademarks of the OpenMP Architecture Review Board

Ubuntu and the Ubuntu logo are registered trademarks of Canonical® Ltd.

Canonical and the Canonical logo are registered trademarks of Canonical Ltd.

OpenCL is a trademark of Apple Inc. used by permission by Khronos Group, Inc.

OpenMPI is distributed by The Open MPI Project under the New BSD license and copyright 2004 - 2012.

AMD 