LUMI Post-upgrade Webinar

What (hasn't) changed in the ROCm stack?

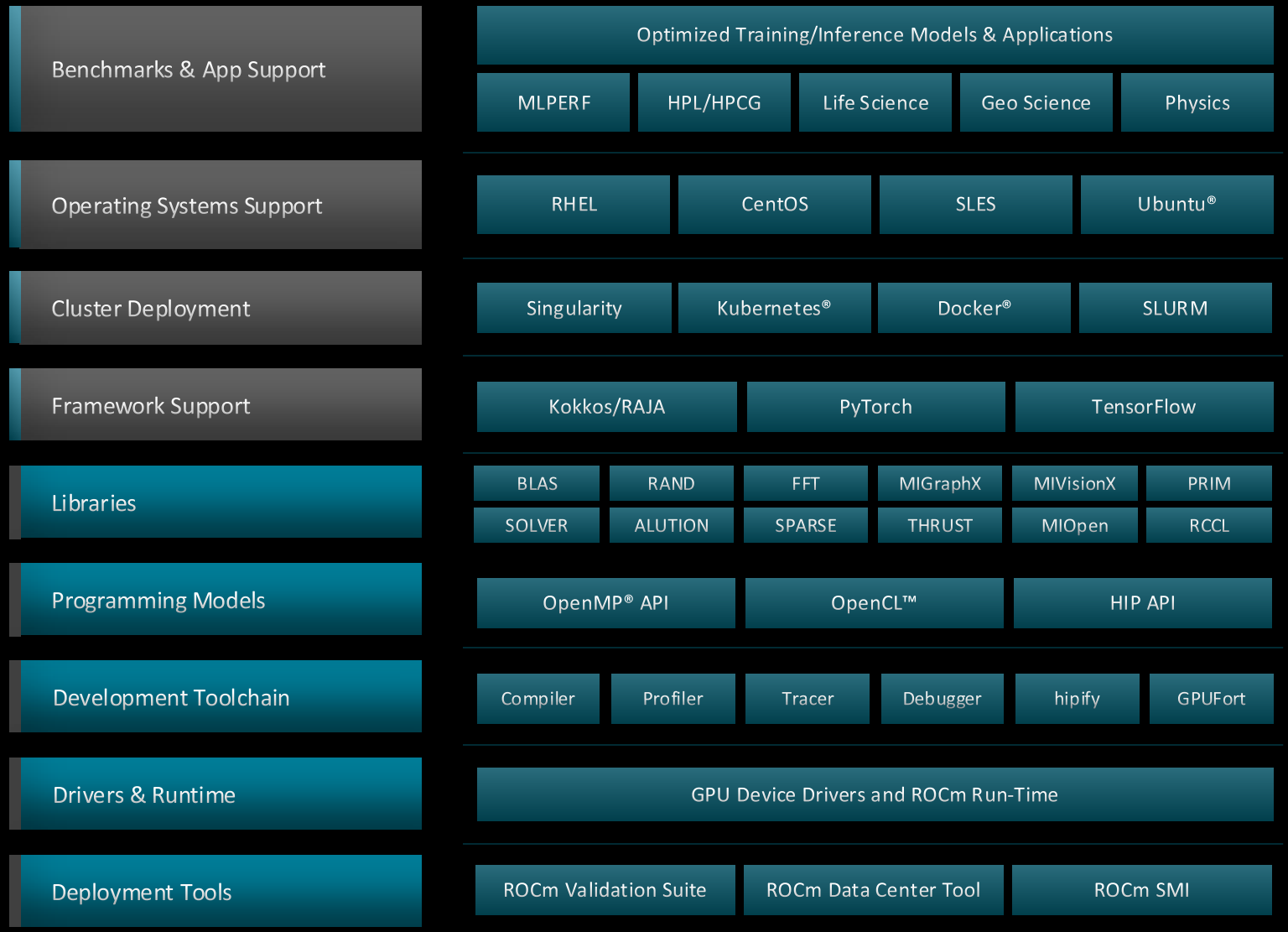
George Markomanolis, Samuel Antao

October 2nd, 2024

ROCm Components Largely the Same as Before!

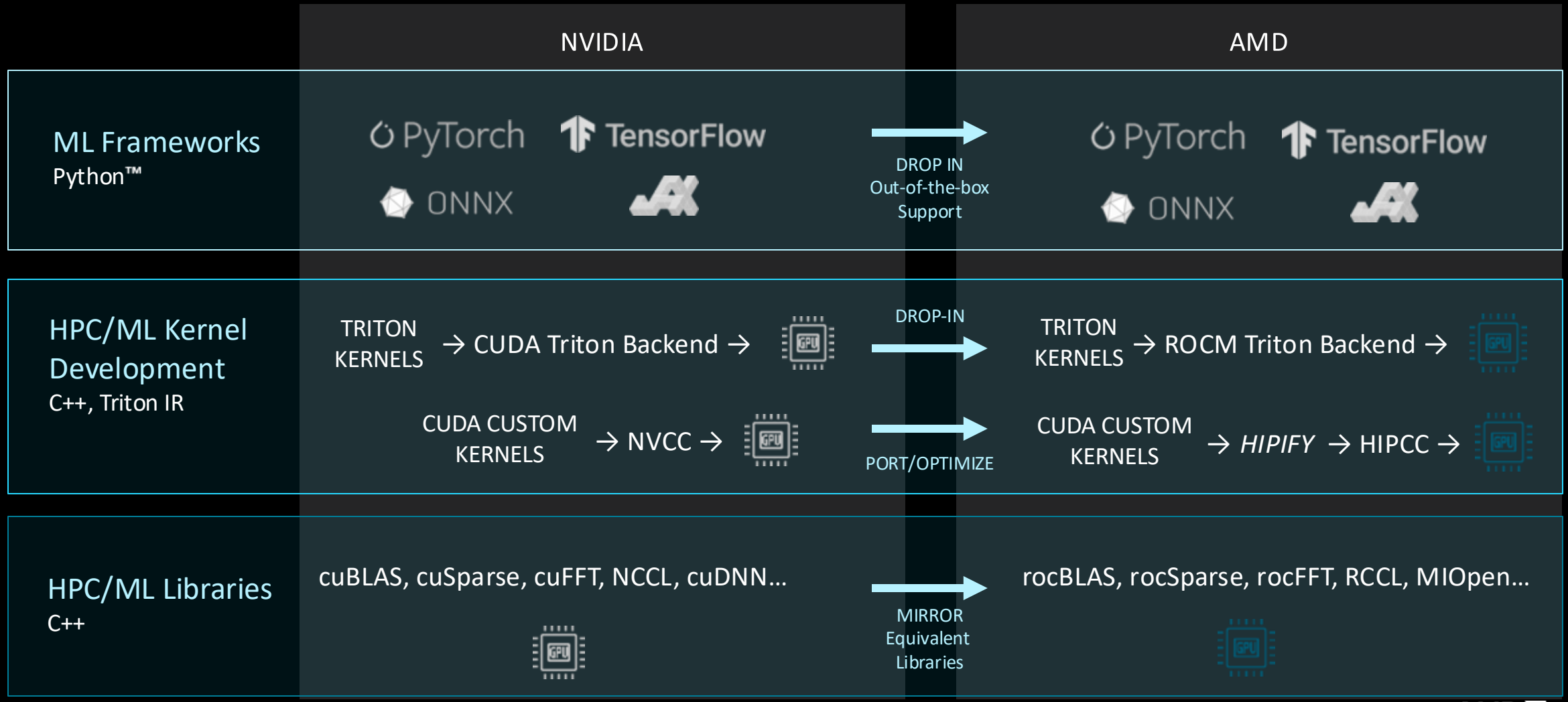
AMD ROCm

- More performance tunning in most components and libraries
- Better support for many community projects in particular around AI
- Expanding the profiling tools
- Better compiler support for Fortran coming very soon!



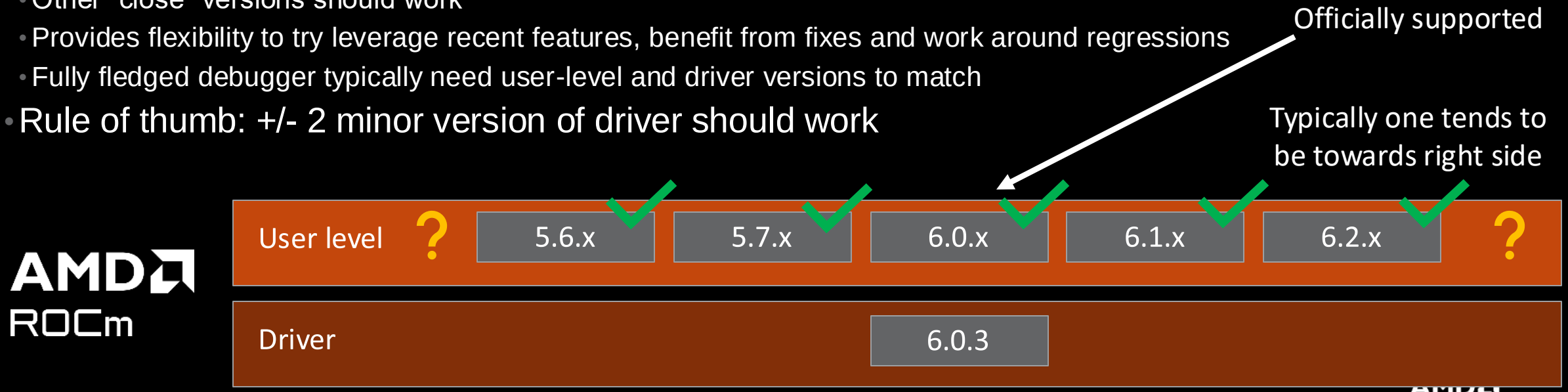
Do I Need to change my porting strategies? No!

PYTHON™ FRAMEWORKS LARGELY DON'T NEED PORTING.



ROCm driver? ROCm User-level? Should I Care?

- ROCm release package includes:
  - Driver support (only changes during scheduled maintenance sessions)
    - Enables the GPU devices being properly referred to in the system
    - Installed by the operations team
  - User-level support (more flexibility to provide newer versions)
    - Tools (compilers/profilers), HIP/HSA runtimes, header files
    - Default version (matching the driver) installed by the operations team
    - Other “close” versions should work
    - Provides flexibility to try leverage recent features, benefit from fixes and work around regressions
    - Fully fledged debugger typically need user-level and driver versions to match
  - Rule of thumb: +/- 2 minor version of driver should work



Containers and module files for recent ROCm versions

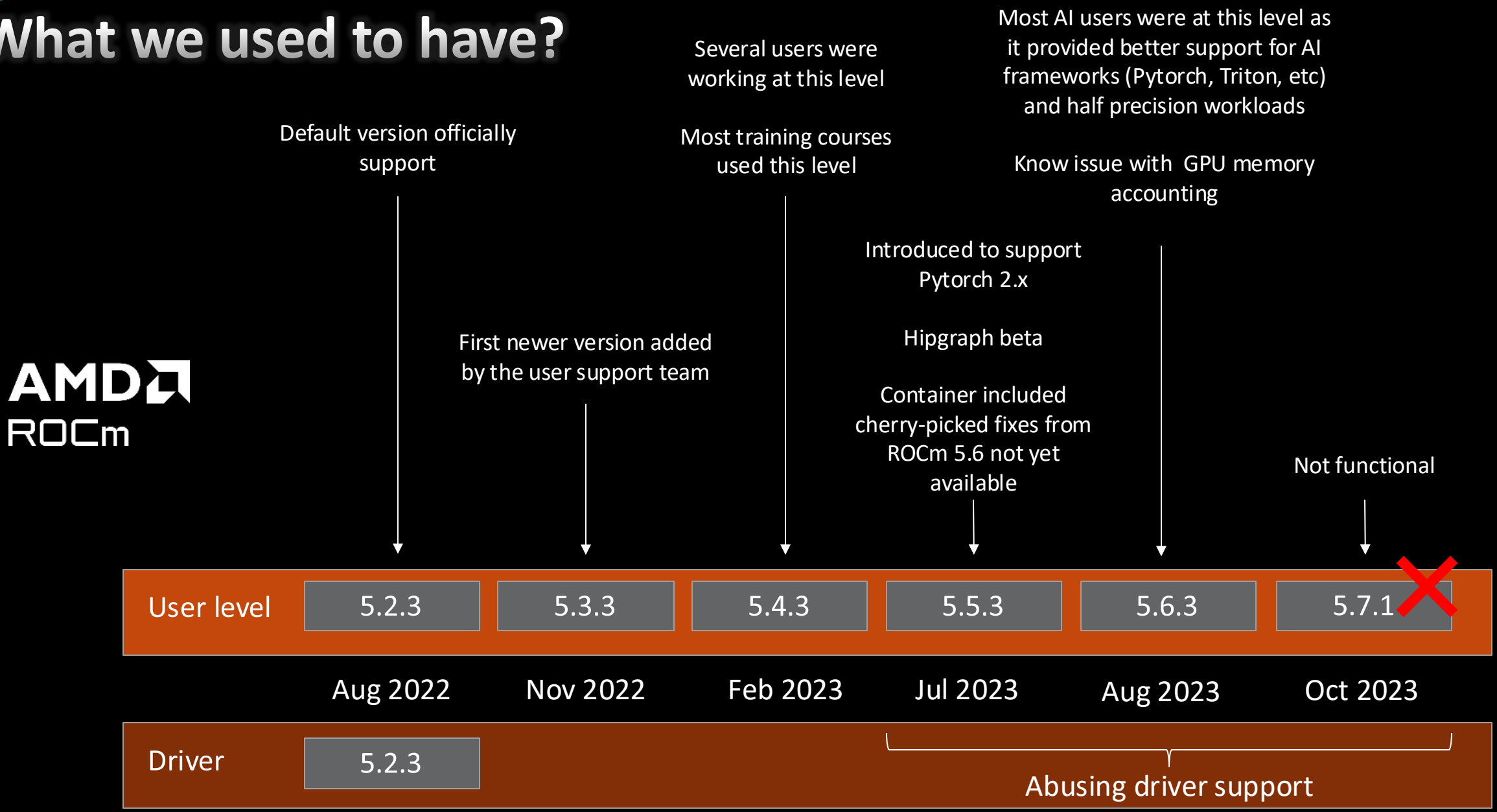
- Historically, newer versions of ROCm have been made available on the system

- Module files
  - ROCm versions 5.2.3 up to 5.6.1
  - Profiler tools (omnitrace and omniperf)
  - RCCL and CXI plug-in for high-performance inter-node comms.
  - `module use /appl/local/containers/test-modules`

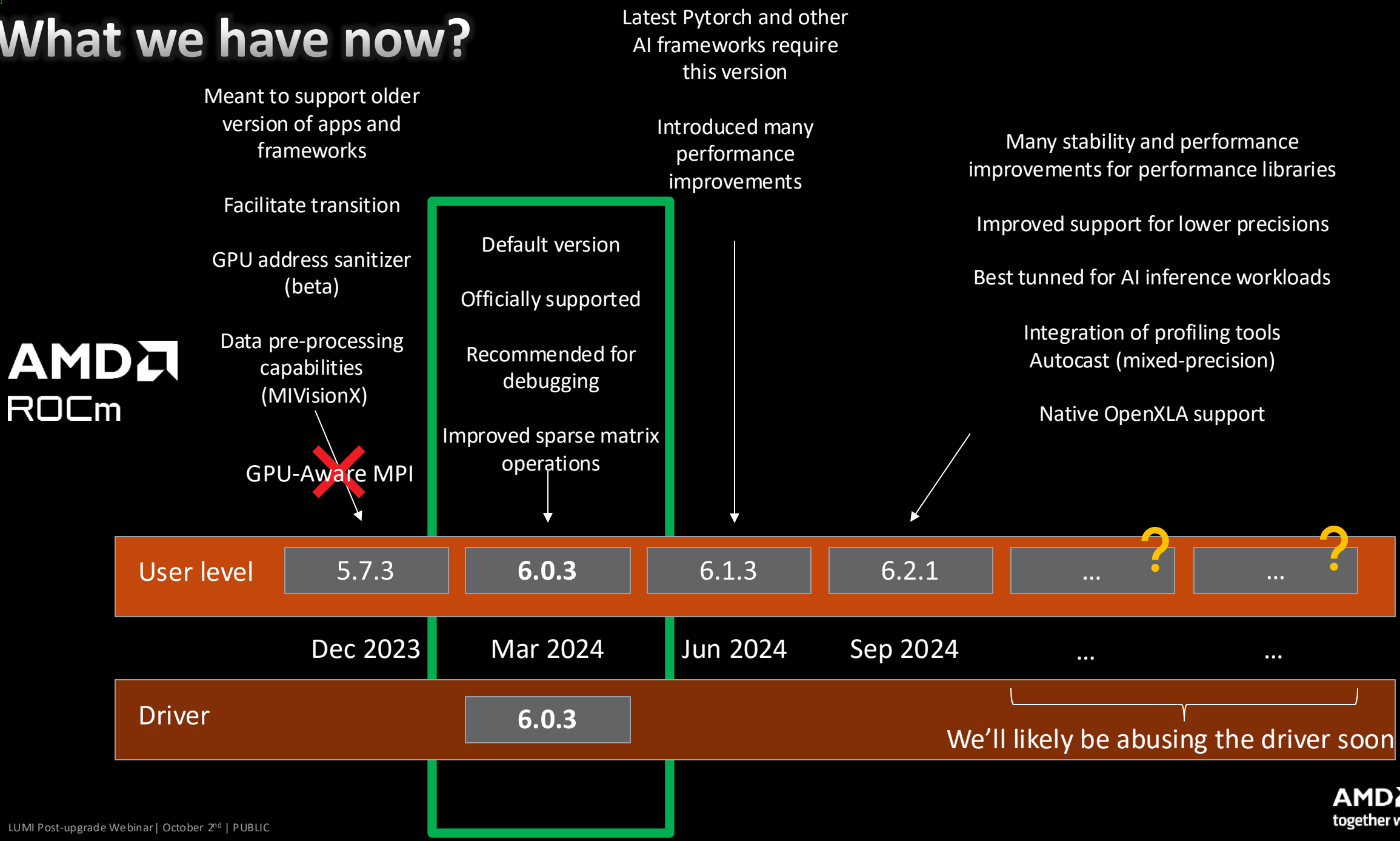
- Containers
  - ROCm versions 5.4.5 up to 5.6.1
  - MPI4Py
  - Pytorch, Tensorflow, JAX
  - Incorporates CXI plugin for better performance while communicating across nodes
  - Recommended way to use Python framework
    - less stress on the filesystem loading your application and several popular packages available
  - Different mechanism available to extend the containers to your taste (Cotainr, local virtual environment, Docker)
  - `singularity exec -B /var/spool/slurmd -B /opt/cray -B /usr/lib64/libcxi.so.1 \`
    `/appl/local/containers/sif-images/lumi-<container name>.sif`

- We expect to continue make newer ROCm versions and supported Python frameworks available
  - ROCm version 5.7.1 up to 6.2.1 (for now)

What we used to have?



What we have now?



ROCm directory structure changes

- Since ROCm 5.3 part of the directory structure was changed and old paths deprecated
- Deprecated directories were removed in ROCm 6
- Can cause build failures for apps doing assumption of the existence of certain directories.

```
/opt/rocm-<ver>
    | --bin
      | --All externally exposed Binaries
    | --libexec
       | --<component>
          | -- Component specific private non-ISA executables (architecture independent)
    | --include
       | -- <component>
          | --<header files>
    | --lib
       | --lib<soname>.so -> lib<soname>.so.major -> lib<soname>.so.major.minor.patch
          (public libraries linked with application)
       | --<component> (component specific private library, executable data)
       | --<cmake>
          | --components
             | --<component>.config.cmake
    | --share
       | --html/<component>/*.html
       | --info/<component>/*.[pdf, md, txt]
       | --man
       | --doc
          | --<component>
             | --<licenses>
       | --<component>
          | --<misc files> (arch independent non-executable)
          | --samples
```

Header files grouped by component, e.g. hipblas.h -> hipblas/hipblas.h

Libraries not grouped by component

Component private libraries (device-tuned kernels) Should be automatically picked up by the relevant libraries

CMake files under /lib
Should not need user intervention and be set properly in the environment

CMake use is encouraged

- CMake support has been featured in ROCm – Make your app build system more robust to changes in the system

```
set(GPU_TARGETS "gfx90a" CACHE INTERNAL "GPU targets to compile for")
set(AMDGPU_TARGETS "gfx90a" CACHE INTERNAL "AMD GPU targets to compile for")
set(CMAKE_HIP_ARCHITECTURES "gfx90a")

mark_as_advanced(GPU_TARGETS)
mark_as_advanced(AMDGPU_TARGETS)
mark_as_advanced(CMAKE_HIP_ARCHITECTURES)


list(APPEND CMAKE_MODULE_PATH ${ROCM_PATH}/lib/cmake/hip)

include(CheckLanguage)
check_language(HIP)

find_package(HIP)
if(HIP_FOUND)
message(STATUS "Found HIP: " ${HIP_VERSION})
else()
message(FATAL_ERROR "Could not find HIP.")
endif()

find_package(hiprand REQUIRED)
find_package(hipblas REQUIRED)
find_package(hipsparse REQUIRED)

target_link_libraries(${devicelib} PRIVATE hip::device)
target_link_libraries(${exec} ${dhostlib} ${devicelib} hip::host hiprand hipblas hipsparse)
```

Select GPU device (gfx90a)

Add the module paths using the new file structure
(the environment might have done this already)

HIP language supported in CMake

Look for specific components

Leverage hip:device target in device libraries linking

Leverage hip:host target in host libraries linking

HIP standard parallelism

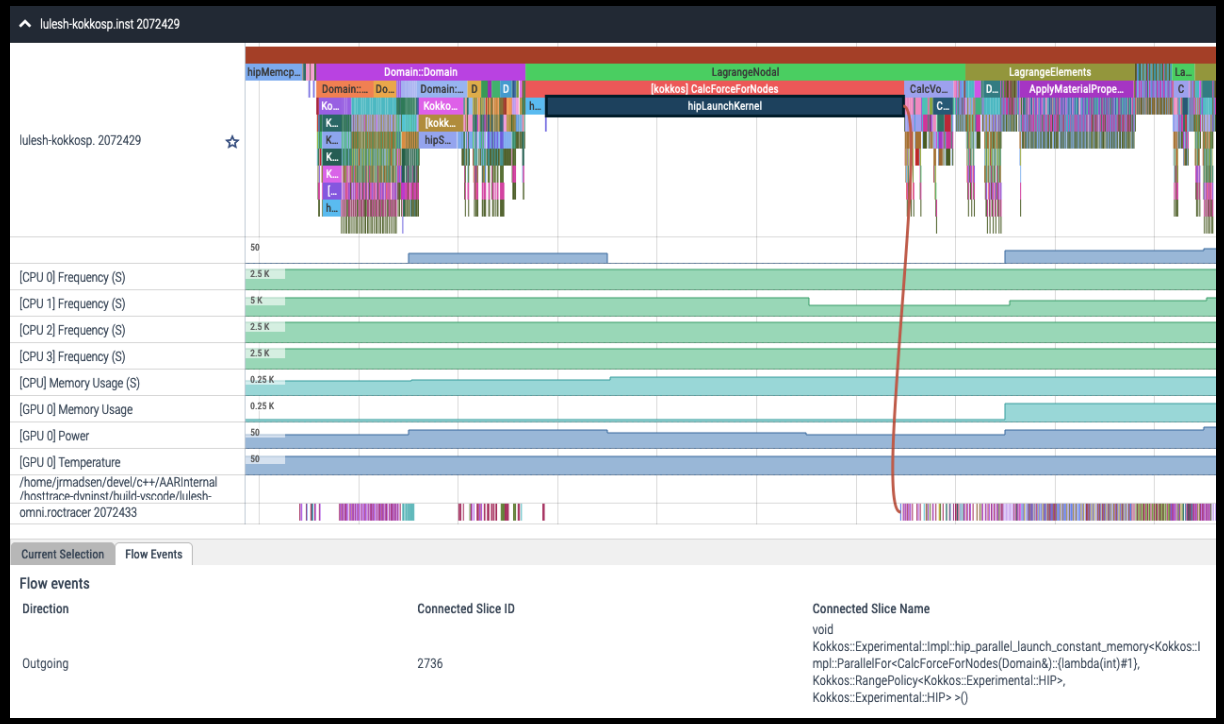
- AMD providing support for advanced C++ in LLVM™ from ROCm 6.2.1
  - Only supports par_unseq acceleration currently
  - Compiler support implementation upstreamed
  - Available in upcoming ROCm releases
  - Re-uses HIP support in CLANG/LLVM and algorithms from libraries (rocThrust)
  - Available today: https://github.com/ROCm/roc-stdpar
  - Several applications tested and running (LULESH, etc.)

```
1  std::transform( // needs <algorithm>
2      std::execution::par_unseq, // <-- needs <execution>
3      indices.begin(), indices.end(), grid.begin(),
4      [](size_t index){
5          return expensive_calculation(index);
6      }
7  );
```

Rocprofiler-SDK and third party tools

- ROCm 6.2 brought several novelties when it comes to profiling

  - Rocprof version 3 available
    - We recommend using this version if possible
    - Traces more compatible with the Perfetto visualizer (use pftrace output)
    - Command options have changed
  - Omnitrace is now part of ROCm distribution
    - New, more stable trace formats
    - Some limitations for Fortran codes
  - Omniperf also part of ROCm distribution
    - Roof-line support for MI200 series GPUs
  - New SDK that can be leveraged by third party tools
  - Several third party tool now enabled on MI200 series GPUs
    - TAU, Score-P, and HPCToolkit are used with AMD GPUs



ROCPROFV3: GETTING STARTED + USEFUL FLAGS

- To get help:
  `${ROCM_PATH}/bin/rocprofv3 -h`
- Useful housekeeping flags:
  - `--hip-trace` For Collecting HIP Traces (runtime + compiler)
  - `--hip-runtime-trace` For Collecting HIP Runtime API Traces
  - `--hip-compiler-trace` For Collecting HIP Compiler generated code Traces
  - `--marker-trace` For Collecting Marker (ROCTx) Traces
  - `--memory-copy-trace` For Collecting Memory Copy Traces
  - `--scratch-memory-trace` For Collecting Scratch Memory operations Traces
  - `--stats` For Collecting statistics of enabled tracing types
  - `--hsa-trace` For Collecting HSA Traces (core + amd + image + finalizer)
  - `--hsa-core-trace` For Collecting HSA API Traces (core API)
  - `--hsa-amd-trace` For Collecting HSA API Traces (AMD-extension API)
  - `--hsa-image-trace` For Collecting HSA API Traces (Image-extenson API)
  - `--hsa-finalizer-trace` For Collecting HSA API Traces (Finalizer-extension API)

ROCPROFV3: GETTING STARTED + USEFUL FLAGS (II)

- Useful housekeeping flags:
  - -s, --sys-trace For Collecting HIP, HSA, Marker (ROCTx), Memory copy, Scratch memory, and Kernel dispatch traces
  - -M, --mangled-kernels Do not demangle the kernel names
  - -T, --truncate-kernels Truncate the demangled kernel names
  - -L, --list-metrics List metrics for counter collection
  - -i INPUT, --input INPUT Input file for counter collection
  - -o OUTPUT_FILE, --output-file OUTPUT_FILE
    For the output file name
  - -d OUTPUT_DIRECTORY, --output-directory OUTPUT_DIRECTORY
    For adding output path where the output files will be saved
  - --output-format {csv,json,pftrace} [{csv,json,pftrace} ...]
    For adding output format (supported formats: csv, json, pftrace)
  - --log-level {fatal,error,warning,info,trace}
    Set the log level
  - --kernel-names KERNEL_NAMES [KERNEL_NAMES ...]
    Filter kernel names
  - --preload [PRELOAD ...]
    Libraries to prepend to LD_PRELOAD (usually for sanitizers)

  - rocprofv3 requires double-hyphen (--) before the application to be executed, e.g.

```
$ rocprofv3 [<rocprofv3-option> ...] -- <application> [<application-arg> ...]
$ rocprofv3 --hip-trace -- ./myapp -n 1
```

  - Instructions: https://rocm.docs.amd.com/projects/rocprofiler-sdk/en/docs-6.2.1/how-to/using-rocprofv3.html

AMD ROCm Developer Hub

Engage with AMD ROCm™ Platform Experts

Participate in AMD ROCm Webinar Series

Post questions, view FAQs in Community Forum

Increase Understanding

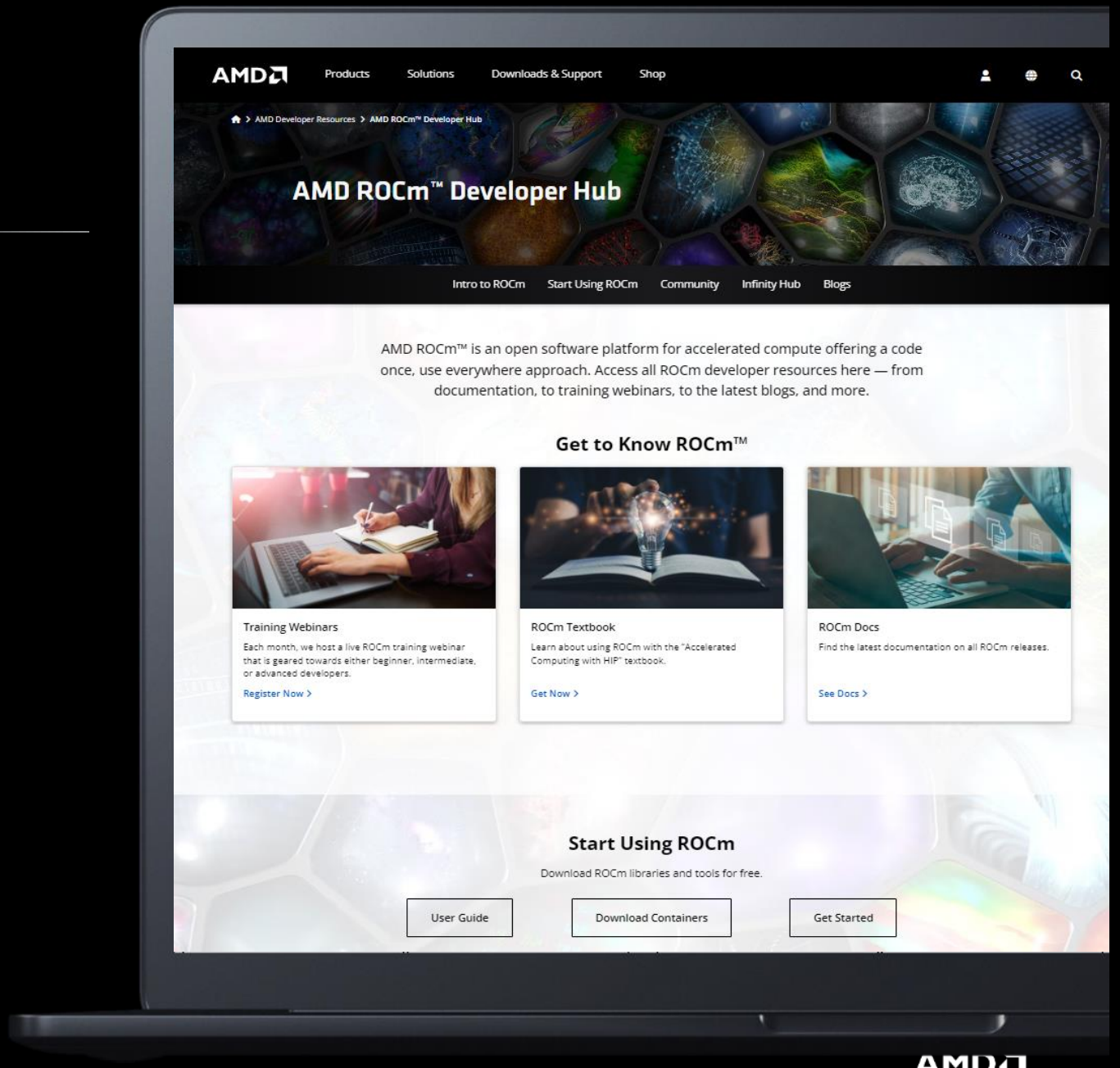
Purchase AMD ROCm Text Book

View the latest news and good practices in the Blogs

Get Started Using AMD ROCm™

AMD ROCm Documentation on GitHub

Download the Latest Version of AMD ROCm



DISCLAIMER AND ATTRIBUTIONS

The information contained herein is for informational purposes only and is subject to change without notice. While every precaution has been taken in the preparation of this document, it may contain technical inaccuracies, omissions and typographical errors, and AMD is under no obligation to update or otherwise correct this information. Advanced Micro Devices, Inc. makes no representations or warranties with respect to the accuracy or completeness of the contents of this document, and assumes no liability of any kind, including the implied warranties of noninfringement, merchantability or fitness for particular purposes, with respect to the operation or use of AMD hardware, software or other products described herein. No license, including implied or arising by estoppel, to any intellectual property rights is granted by this document. Terms and limitations applicable to the purchase or use of AMD products are as set forth in a signed agreement between the parties or in AMD's Standard Terms and Conditions of Sale. GD-18