

The background of the slide is a solid blue color. In the upper right corner, there is a faint, glowing circular graphic that resembles a stylized eye or a complex lens, composed of concentric rings and radial lines. In the upper left, there are some faint, abstract shapes that look like stylized arrows or data paths. The main title is centered in the upper half of the slide in a large, white, sans-serif font.

HPCToolkit + Rocprofiler-sdk Working Group

John Mellor-Crummey, Jonathon Madsen, David Boehme
Scalable Tools Workshop 2026

Topics

- Deadlock at termination when collecting PC samples
- Problems with tool compatibility across ROCm versions

Investigating Deadlock at Termination when PC Sampling

- **Running GpuPerfSuite kernels multiple times causes deadlock when flushing PC samples at application termination**
 - AMD's flushing code waits forever for a signal that never gets delivered
 - Hypothesis: improper termination in GpuPerfSuite draft benchmark code
 - didn't contain a hipDeviceSynchronize before calling exit
 - teardown of GPU runtime at exit may cause signal to be lost
 - Recommendation: call hipDeviceSynchronize before calling exit
 - Follow-up
 - adding hipDeviceSynchronize before termination didn't help
- **Status: unresolved deadlock**

Deadlock at Termination when PC Sampling

- #0 0x00007ffff4ef422b in ioctl () from /lib64/libc.so.6
- #1 0x00007fffea8cd020 in hsakmt_ioctl () from /opt/rocm-7.1.1/lib/libhsa-runtime64.so.1
- #2 0x00007fffea8c3dc3 in hsaKmtWaitOnMultipleEvents_Ext () from /opt/rocm-7.1.1/lib/libhsa-runtime64.so.1
- #3 0x00007fffea8c4537 in hsaKmtWaitOnEvent_Ext () from /opt/rocm-7.1.1/lib/libhsa-runtime64.so.1
- #4 0x00007fffea815b87 in rocr::core::InterruptSignal::WaitRelaxed(hsa_signal_condition_t, long, unsigned long, hsa_wait_state_t) ()
from /opt/rocm-7.1.1/lib/libhsa-runtime64.so.1
- #5 0x00007fffea81588a in rocr::core::InterruptSignal::WaitAcquire(hsa_signal_condition_t, long, unsigned long, hsa_wait_state_t) ()
from /opt/rocm-7.1.1/lib/libhsa-runtime64.so.1
- #6 0x00007fffea809a41 in rocr::HSA::hsa_signal_wait_scacquire(hsa_signal_s, hsa_signal_condition_t, long, unsigned long, hsa_wait_state_t) ()
from /opt/rocm-7.1.1/lib/libhsa-runtime64.so.1
- #7 0x00007fffea7ee2f6 in rocr::AMD::GpuAgent::PcSamplingFlushDeviceBuffers(rocr::pcs::PcsRuntime::PcSamplingSession&) ()
from /opt/rocm-7.1.1/lib/libhsa-runtime64.so.1
- #8 0x00007fffea7ee534 in rocr::AMD::GpuAgent::PcSamplingFlush(rocr::pcs::PcsRuntime::PcSamplingSession&) ()
from /opt/rocm-7.1.1/lib/libhsa-runtime64.so.1
- #9 0x00007fffea88e4f5 in rocr::pcs::PcsRuntime::PcSamplingFlush(hsa_ven_amd_pcs_t) () from /opt/rocm-7.1.1/lib/libhsa-runtime64.so.1
- #10 0x00007fffea88d7c8 in rocr::pcs::hsa_ven_amd_pcs_flush(hsa_ven_amd_pcs_t) () from /opt/rocm-7.1.1/lib/libhsa-runtime64.so.1
- #11 0x00007fffe6ffda02 in std::_Function_handler<void (), rocrprofiler::pc_sampling::hsa::flush_internal_agent_buffers(rocrprofiler::pc_sampling::PCSAgentSession const*)>::lambda(#1) >::_M_invoke(std::_Any_data const&) () from /opt/rocm-7.1.1/lib/librocrprofiler-sdk.so.1
- #12 0x00007fffe6ffedd3 in rocrprofiler::pc_sampling::PCSCIDManager::manage_cids_explicit(std::function<void ()> const&) ()
from /opt/rocm-7.1.1/lib/librocrprofiler-sdk.so.1
- #13 0x00007fffe6ffd2a0 in rocrprofiler::pc_sampling::hsa::flush_internal_agent_buffers(rocrprofiler::pc_sampling::PCSAgentSession const*) ()
from /opt/rocm-7.1.1/lib/librocrprofiler-sdk.so.1
- #14 0x00007fffe6ffdf8f in rocrprofiler::pc_sampling::flush_internal_agent_buffers(rocrprofiler_buffer_id_t) ()
from /opt/rocm-7.1.1/lib/librocrprofiler-sdk.so.1
- #15 0x00007fffe6af66b9 in rocrprofiler_flush_buffer () from /opt/rocm-7.1.1/lib/librocrprofiler-sdk.so.1
- ...

Investigating Tool Compatibility Across ROCm Versions

- **Problem**

- Rocprofiler-sdk has been evolving rapidly
- Difficulties building a tool with one ROCm minor version and using it on programs compiled with a different minor version

- **Wanted**

- Tool backward compatibility
 - Build tool with newest Rocprofiler-sdk and have it work with earlier ROCm versions
- Tool forward compatibility across minor versions
 - Build tool with one ROCm minor version and have it work for later minor versions

Forward Compatibility Investigation

- **Experiment**

- compile HPCToolkit against ROCm 7.1.1
- monitor program compiled with ROCm 7.2.4

- **Result**

- Bad AQL packet causes SEGV in AMD ROCm runtime
- Noted that AQL profiling library was being loaded from 7.1.1 (tool version) when running program compiled with 7.2.4

- **Concern**

- later runtime may need later AQL packet library
- Rocprofiler-sdk 7.1.1 loads 7.1.1 libhsa-amd-aqlprofile64 library

- **Status: unresolved understanding issues affecting forward compatibility across minor versions**

- unclear why GpuPerfSuite collecting PC samples and tracing GPU operations needs AQL
- some improvements separating AQL from Rocprofiler-sdk/ROCm versions
- cross-version compatibility needs more work

Forward Compatibility Investigation

- **GpuPerfSuite issues when using HPCToolkit + stochastic sampling**

- non-deterministic errors
 - HIP error: unspecified launch failure
 - Queue at 0x7f20e44f8000 inactivated due to async error:
HSA_STATUS_ERROR_INVALID_PACKET_FORMAT: The AQL packet is malformed.

- **Libraries loaded**

- /opt/rocm-7.1.1/lib/libhsa-amd-aqlprofile64.so.1.0.70101
- /opt/rocm-7.1.1/lib/librocprofiler-sdk.so.1.0.0
- /opt/rocm-7.2.4/lib/libamd_comgr.so.3.0.0
- /opt/rocm-7.2.4/lib/libamdhip64.so.7.2.70204
- /opt/rocm-7.2.4/lib/libhsa-runtime64.so.1.18.70204
- /opt/rocm-7.2.4/lib/librocprofiler-register.so.0.6.0
- /storage/users/johnmc/hpctoolkit/hpctoolkit-build-7.1/src/hpcrun/libhpcrun_dlopen_rocm.so
- /storage/users/johnmc/hpctoolkit/hpctoolkit-build-7.1/src/hpcrun/libhpcrun_preload_rocm.so

Backward Compatibility

- **Not investigated today**
- **Problems observed in the past across ROCm versions with PC sampling**
 - Across minor versions in ROCm 6
 - Between ROCm 7 and ROCm 6
- **John M-C doesn't have an understanding of current status**