Game optimization
Radeon™ developer tools on RADV & Steam Deck™

Adam Sawicki, AMD    Friedrich Vock, Valve    Timur Kristóf, Valve
# Table of Contents

1. About RADV and Steam Deck  
2. Intro to Radeon Developer Tools  
3. Radeon GPU Profiler  
4. Radeon Raytracing Analyzer  
5. Radeon Memory Visualizer
About RADV
What is RADV?

RADV is a Vulkan® driver for AMD Radeon GPUs that is written for Linux® by the community.

Supported GPUs
All GCN & RDNA™ starting with GCN 1, up to RDNA3.
What RADV isn’t

- NOT a kernel driver (that is AMDGPU)
- NOT an OpenGL® driver (that is RadeonSI)
- NOT officially developed by AMD
Why we work on RADV

Open development model
RADV is open source (without closed parts). Development happens in the upstream Mesa repo, which makes it transparent and easier for new contributors to join.

Benefits of sharing code
RADV shares code with other drivers in the Mesa project, benefiting from the work of developers who work on those drivers.

Flexibility
Direction and future plans are community-driven and discussed openly. We can work on features and bugfixes without relying solely on the vendor.
Unique features of RADV

• ACO, a custom shader compiler optimized for gaming
• Supports several Vulkan extensions for API layering purposes
About Steam Deck
Steam Deck
Steam Deck API situation

Proprietary API use is translated to open standard based APIs using translation layers.

- D3D12 games - Vulkan (VKD3D-Proton)
- D3D9-11 games - Vulkan (DXVK)
- older D3D games - OpenGL (WineD3D)
Steam Deck driver situation

Graphics drivers come from the Mesa project.

- Vulkan - RADV
- OpenGL - RadeonSI
Intro to Radeon Developer Tools
• https://gpuopen.com/tools/
• Freely available tool suite for AMD GPUs
• Windows 10/11, Linux, Vulkan, DX12, and more...
• Connects to the graphics driver, captures low-level information
Radeon GPU Profiler (RGP)
• Captures all GPU work spent on a single frame
• Correlates detailed hardware traces (SQTT) and streaming counters (SPM) to driver and app data
• Wavefront occupancy timeline
• Shader disassembly and instruction timing
• Pipeline info, context rolls, barriers, etc.
• Meshes/textures are not captured
Radeon GPU Profiler (RGP)
RADV RGP integration

- RADV enables SQTT/SPM
- SQTT/SPM collects perf data in VRAM
- RADV copies the data from VRAM to a file
RADV RGP integration

How to use on Deck:

• “RGP Capture” button in the SteamOS Devkit Client

How to use on Linux:

• RADV_THREAD_TRACE=frameindex
• RADV_THREAD_TRACE_TRIGGER=filename
Radeon GPU Profiler (RGP)

Demo.
Radeon Raytracing Analyzer (RRA)
• Visualization and analysis of Raytracing Acceleration Structures (BVH)
• Traversal counter heatmap
• Useful for both debugging and profiling
Radeon Raytracing Analyzer (RRA)
RADV RRA integration

- Traces contain built acceleration structures in AMD-official driver format
- RADV has to convert BVH data to the AMD format
- After acceleration structures are built, they are copied to host memory in order to be written to the trace file
- Conversion happens on the host
- Added in Mesa 22.3
RADV RRA integration

How to use:

- `RADV_RRA_TRACE=frameindex`
- `RADV_RRA_TRACE_TRIGGER=filename`
- Optional validation during conversion: `RADV_RRA_TRACE_VALIDATE=1`
Radeon Raytracing Analyzer (RRA)

Demo.
Radeon Memory Visualizer (RMV)
• Analyzes all allocated and deallocated resources created by the app over its entire lifetime
• Driver-internal memory consumption is included as well
• Instruments the kernel-mode driver for residency information
• Memory state at different points in time can be compared, for example to diagnose memory leaks
RADV RMV integration

• Allocations and resources are captured inside RADV
• Residency events are written by the kernel to a ring buffer (needs Linux 5.10 or newer)
• RADV reads the ring buffer and logs residency events
• Added in Mesa 23.0
RADV RMV integration

How to use:

- Ring buffer needs to be configured by the user through the `setup.sh` script shipped with the Radeon Developer Tool Suite
- `MESA_VK_MEMORY_TRACE=frameindex`
- `MESA_VK_MEMORY_TRACE_TRIGGER=filename`
Radeon Memory Visualizer (RMV)

Demo.
Thanks

Alexander Deucher
Christopher Hesik
Christian König
Marek Olšák
Pierre-Eric Pelloux-Prayer
Budi Purnomo

Bas Nieuwenhuizen
Dave Airlie
Samuel Pitoiset
Daniel Schürmann
Rhys Perry
Timur Kristóf
Tatsuyuki Ishi
Georg Lehmann
Konstantin Seurer
Friedrich Vock
Game optimization
Radeon™ developer tools on RADV & Steam Deck™

Adam Sawicki, AMD  Friedrich Vock, Valve  Timur Kristóf, Valve