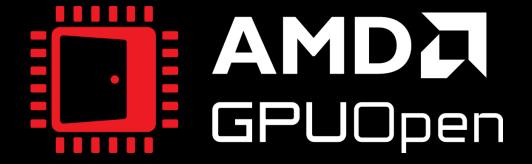
AMD

OPTIMIZING GAME PERFORMANCE WITH THE RADEON[™] DEVELOPER TOOL SUITE

CHRIS HESIK

CAN ALPER



AMD together we advance_

ABOUT THIS PRESENTATION

Part 1: Overview of the Radeon[™] Developer Tool Suite

- What is in it
- What's new since GDC 2022
- Dive into new features and improvements
- RADV & Steam Deck support

Part 2: Collaboration with external tool developers

- Microsoft® PIX on Windows
- RenderDoc
- GFXReconstruct



AMDJ RADEON Developer Tool Suite

AMDJAMDJAMDJAMDJAMDJAMDJRADEON
GPU ProfilerRADEON
Memory VisualizerRADEON
Raytracing AnalyzerRADD
CPU AnalyzerRADD
CPU AnalyzerRADD
CPU Analyzer



Radeon GPU Profiler (RGP)

- Get a bird's eye view of how your command buffers got submitted to each GPU queue, including synchronization between the queues
- Understand how your wavefronts were pushed through the GPU, and correlate wavefronts to the API-level Draws/Dispatches which launched them
- Learn how your frame utilizes the various GPU memory caches
- Quickly and easily find hotspots in your shaders using the instruction timing view
- Identify parts of the frame using ray tracing and gain insights into the different shaders that make up a ray tracing pipeline, including which parts may be the bottleneck
- Find out which barriers flushed caches, caused a synchronization point or even ran their own, internal shaders
- Discover which events and which pipelines are the most expensive.
- Get insight into the render targets used by your frame

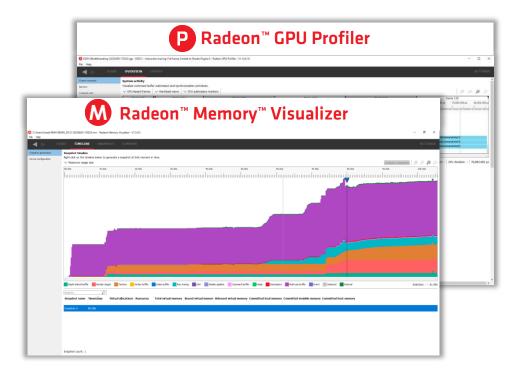
Help	9-174532/gp - 03012 - Instruction tracing: Full frame, limited to Shader Engine D - Radeon GPU Prefile - V1.14.0.10			- 0
t In stant	OVERVIEW EVENIN			settine
n na manag	System activity Visualae common buffer submission and synchronization primitives. V OPU-based frames: V Workbod views: V OPU-submission markers			[# # # #
esperaise exects ecoleptic targets nes e configuration	Pare 119 Pere 121 Fine 222 A005 A 1,000 Opt 1,000 Opt 1,000 Opt 20,000 Opt 1,000 Opt 1	10.001.001 pr 35,000.000 pr 40,000.000 pr 41,001.000 pr		Prame 123 45,000.000 yr 75,000.000 yr 75,000.000 yr 80,000.000 rafar yr affaranau fa maeffaran af armenfa
	EXCODes desinformation EXCODes des des des des des des des des des d	E0012para (Encodermanilate) () E0012para (Encodermanilate) () E0012para (Encodermanilate) () E0012para (Encodermanilate) () E0012para (Encodermanilate) () E0012para (Encodermanilate) ()	DD 12pare: Securitizerentative() DD 12pare: Securitizerentativ() DD 12pare: Securitizerentativ() DD 12pare: Securitizerentativ() DD 12pare: Securitizerentativ() DD 12pare: Securitizerentativ()	100 100 sei al Energia Commendia (a) 100 100 sei a Energia Commendia (a) 100 100 sei a Energia Commendia (a)
	c Graphics Computer DMA		Submit time: - Submit	hration: - Enginee duration: - GPU duration: - 76,803.603
		3 tot	Set the - Set of F	and the second s





Radeon Memory Visualizer (RMV)

- Tracks the GPU memory usage of a running application
- Create detailed snapshots of memory state
- Visualize heap over-subscription
- Quickly find resources which are not in the optimal heaps
- Understand which resources require the most memory, which heaps they are in, and how your heaps are sub-allocated
- Identify fragmentation and understand how you are managing memory in each heap
- Understand the balance between dedicated and placed resources
- Easily find memory leaks in your application by comparing snapshots
- Use the DirectX[®] 12 or Vulkan[®] API to customize the names of your resources then search and view the details in the Resource list

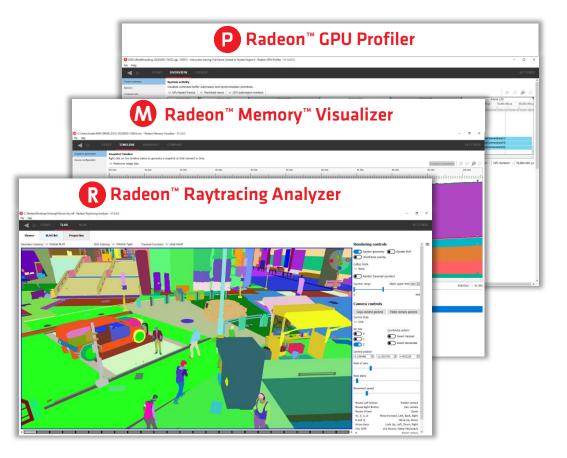






Radeon Raytracing Analyzer (RRA)

- Visually inspect your acceleration structures (TLAS and BLAS), select coloring modes to highlight areas of interest
- Use traversal counter rendering mode to find hotspots for loop count, instance overlaps and more
- View and validate primitives and attributes that were passed to the driver by your application
- High-level statistics for every acceleration structure in your scene, including things such as memory usage and triangle counts

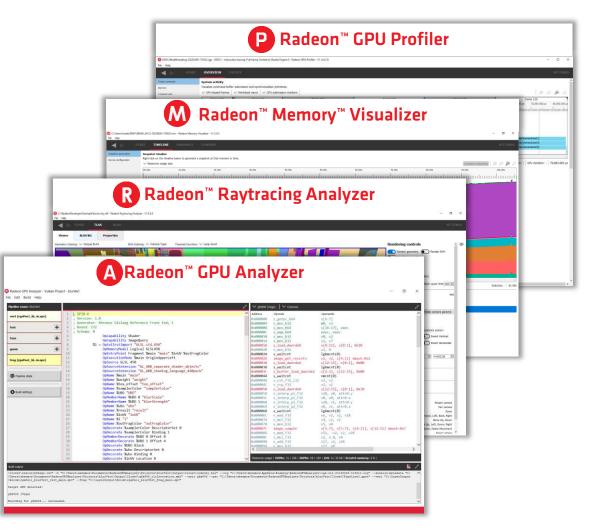






Radeon GPU Analyzer (RGA)

- Compile and analyze shaders and kernels for AMD GPUs
- Inspect disassembly and static hardware resource usage
- Identify areas with high VGPR pressure
- Independent of the physically installed GPU
- CLI support for DirectX[®]12, DXR, DirectX[®]11, Vulkan[®], SPIR-V[™], OpenGL[®], and OpenCL[™]
- GUI support for Vulkan and OpenCL







D Radeon [™] Develope	r Pa	nel		
O Local - Radeon Developer Panel v2.6.0.0			- 0	×
CONNECTION SYSTEM APPLICATIONS				?
D3D12RaytracingSi Profiling Raytracing Memory Trace Device Cl	locks			
Status: Online				
Capture scene				
Hotke Ctrl+Alt+L				
Recently collected scenes				
Path: D:\Development\rra_scenes\D3D12Raytra	acingSimp	leLighting		
Scene	Size	Created		^
D3D12RaytracingSimpleLighting-20220121-114545.m			11:45:45 2022	
D3D12RaytracingSimpleLighting-20220121-103146.rr D3D12RaytracingSimpleLighting-20220121-101710.rr				
Open				
<pre></pre>				
Game Annlicati	inn			

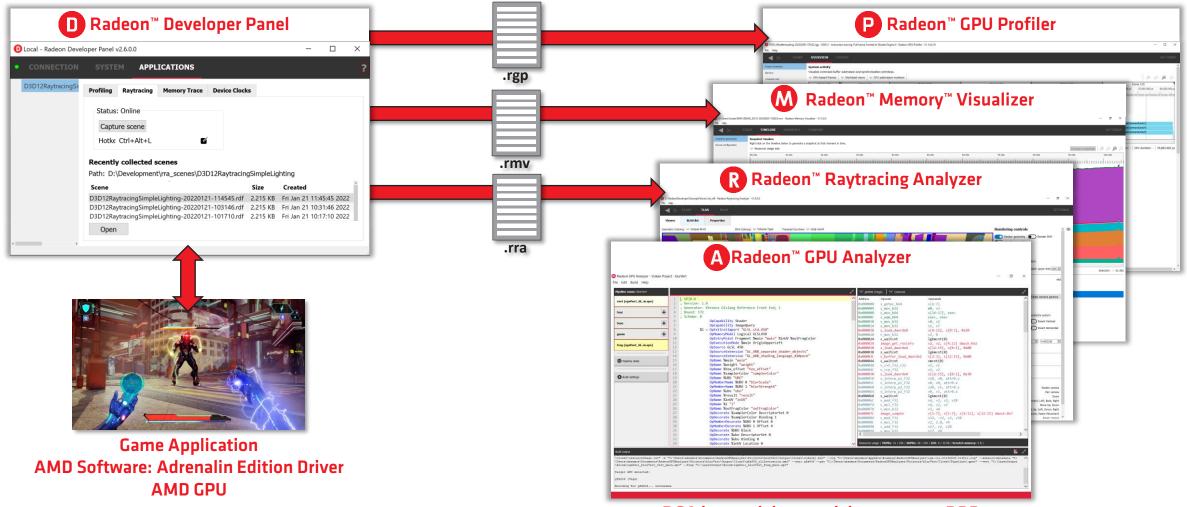
Game Application AMD Software: Adrenalin Edition Driver AMD GPU

Radeon Developer Panel (RDP)

- RDP connects to the Radeon[™] Adrenalin driver to enable developer mode
- Developer mode generates:
 - Correlated hardware tracing, driver and application timing data for the Radeon[™] GPU Profiler (RGP)
 - Memory usage data for the Radeon[™] Memory Visualizer (RMV)
 - Acceleration structure information for RRA
- Supports local and remote data gathering
- RDP supports default and custom workflows to simplify data gathering from your applications







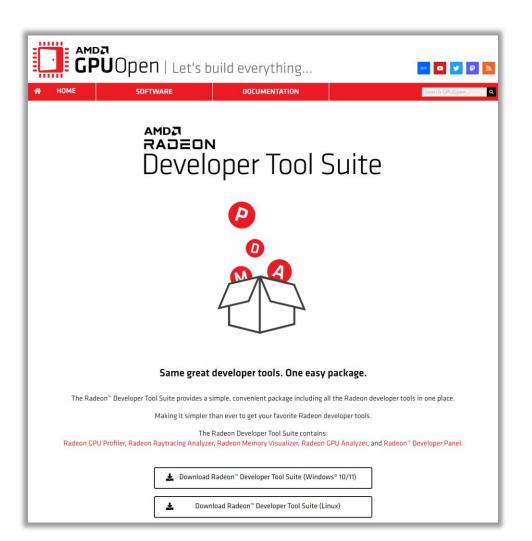
RGA is standalone and does not use RDP





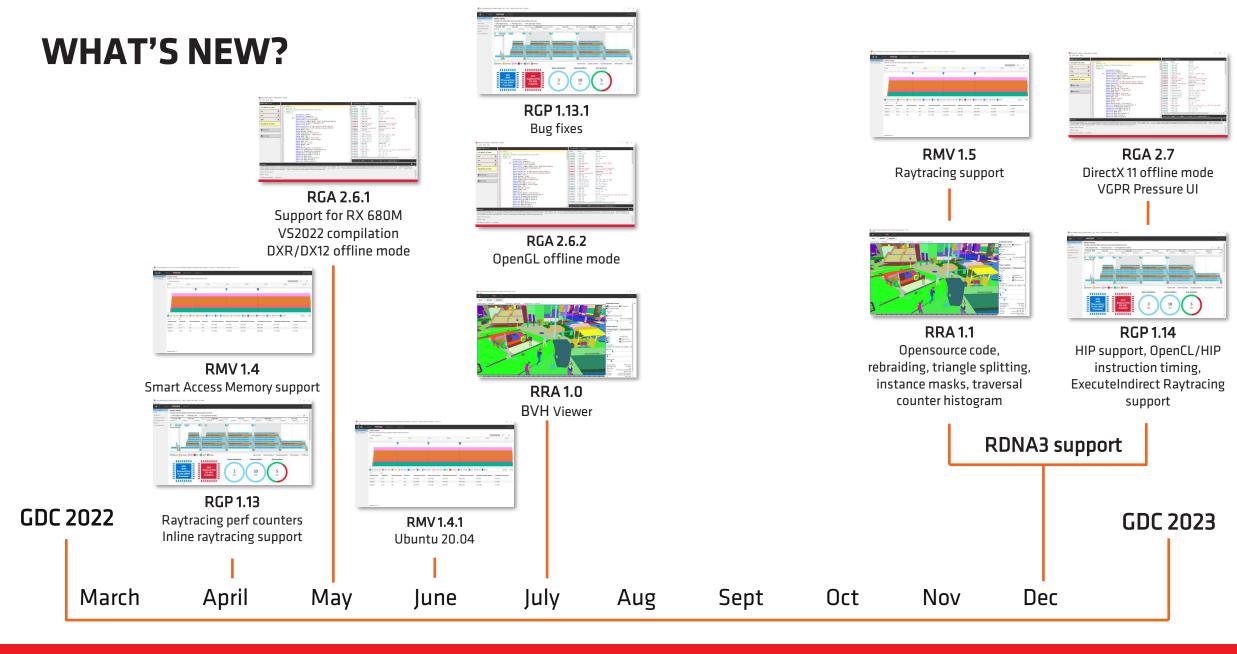
RADEON DEVELOPER TOOL SUITE

- Download: gpuopen.com/tools
- ZIP file containing
 - Radeon GPU Profiler
 - Radeon Memory Visualizer
 - Radeon Raytracing Analyzer
 - Radeon GPU Analyzer
 - Radeon Developer Panel
- No installer needed
- Videos: gpuopen.com/videos











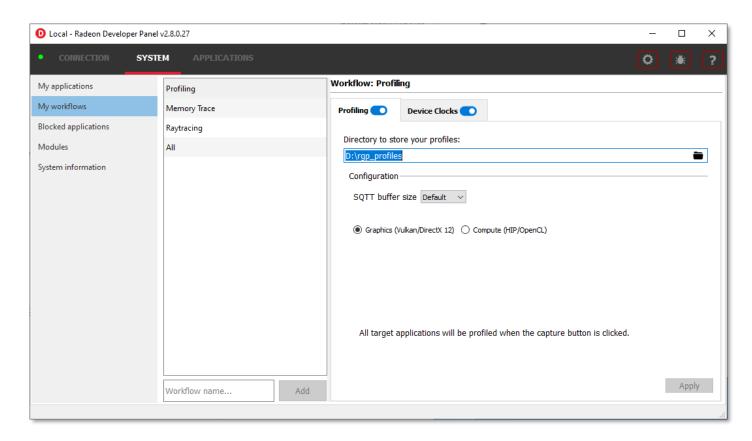


AMDJ RADEON Developer Panel



RADEON DEVELOPER PANEL 2.8 UPDATE

• Workflow UI improvements







Developer Panel

RADEON DEVELOPER PANEL 2.8 UPDATE

- Workflow UI improvements
- New System Information

D Local - Radeon Developer Panel	l v2.8.0.27		_		×
CONNECTION SYST	EM APPLICATIONS		٥	☀	?
My applications					^
My workflows	Host System				
	OS Name:	Windows 10 Pro			
Blocked applications	OS Description:	19041.1.amd64fre.vb_release.191206-1406			
Modules	Hostname:	DESKTOP-AA9V7RI			
System information	Physical memory:	63.949 GB			
-,	Swap memory:	73.449 GB			
	Driver				
	Name:	AMD Windows			
	Description:	AMD Windows Driver			
	Packaging version:	22.20.29.04-221025a-384991E-AMD-Software-Adrenalin-Edition			
	Software version:	22.10.3			
	СРИ О				
	Name:	AMD Ryzen 7 3700X 8-Core Processor			
	Architecture:	x64			
	Vendor ID:	AuthenticAMD			
	CPU ID:	AMD64 Family 23 Model 113 Stepping 0			
	Device ID:	CPU0			
	Physical core count:	8			
	Logical core count:	16			
	Speed:	3.59 GHz			
	Virtualization:	disabled			
	GPU O				
	Name:	AMD Radeon RX 6900 XT			
	Shader engine clock frequency (min):	500 MHz			
	Shader engine clock frequency (max):	: 2250 MHz			
	Timestamp frequency:	100 MHz			
	Family:	8F			
	Device ID:	73AF			





Developer Panel

RADEON DEVELOPER PANEL 2.8 UPDATE

- Workflow UI improvements
- New System Information
 - With Export (JSON)
- RDNA3 Support
- Stability

Device ID:	CPUU
Physical core count:	8
Logical core count:	16
Speed:	3.59 GHz
Virtualization:	disabled
GPU O	
Name:	AMD Radeon RX 6900 XT
Shader engine clock frequency (min):	500 MHz
Shader engine clock frequency (max)	: 2250 MHz
Timestamp frequency:	100 MHz
Family:	8F
Device ID:	73AF
Revision:	C0
eRev:	29
Memory	
Bandwidth:	608 GB/s
Bus bit width:	256
Clock frequency (min):	96 MHz
Clock frequency (max):	1188 MHz
Operations per clock:	16
Invisible heap size:	15.734 GB
Local heap size:	256.000 MB
PCI	
Bus:	11
Device:	0
Function:	0
Big SW	
Major:	2021
Minor:	1
Misc:	0
	¥
	Export
	· · · ·





AMDJ RADEON GPU Profiler

https://github.com/GPUOpen-Tools/radeon_gpu_profiler





RADEON GPU PROFILER 1.14 – RDNA 3 SUPPORT

▲ ►s	TART OVERVIEW EVENTS		SETTI	
Frame summary				
Barriers	AM			
Context rolls				
Most expensive events	System information			
Render/depth targets	Processor name: Clock speed:	AMD Ryzen 9 5950X 16-Core Processor 3394 MHz		
Pipelines	Physical cores:	16		
Device configuration	Logical cores:	32		
evice configuration	System memory (RAM):	128 GB		
	GPU information			
	Device name:	AMD Radeon RX 7900 XTX		
	Device ID (and revision):	744CC8		
	Shader core			
	Shader core clock frequency: Shader engines:	2269 MHz (2268 MHz peak) 6		
	Work group processors per shader engine:	8		
	SIMD per work group processor:	4		
	Wavefronts per SIMD:	16		
	Vector registers per SIMD: Scalar registers per SIMD:	1536 2048		
		2010		
	Memory			
	Video memory clock frequency: Video memory bandwidth:	1250 MHz (1250 MHz peak) 960.0 GB/s		
	Video memory size:	24 GB		
	Video memory type:	GDDR6		
	L0 vector cache size per compute unit:	16 KB		
	L1 cache size per shader array:	256 KB <u>4 MB</u>		
	Infinity cache size:	96 MB		
	Instruction cache size per compute unit:	32 KB		
	Scalar cache size per compute unit LDS size per work group processor:	16 KB 128 KB		
	LDD size per work group processor.	120 10		

- Additional cache size info in the Device Configuration pane
- More Vector registers per SIMD



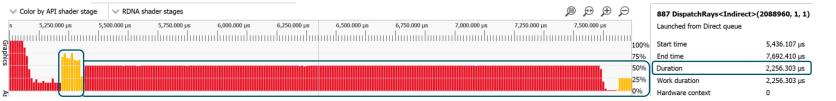


SPLL Profile

RADEON GPU PROFILER 1.14 – RDNA 3 SUPPORT

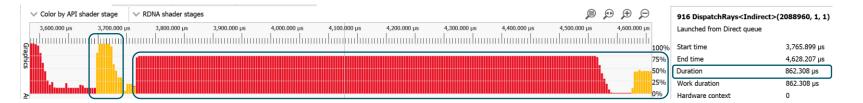
AMD Radeon RX 6800

AMD Ryzen 7 5800X 8-Core Processor, 3793 MHz, 64GB RAM, Microsoft Windows 11 Pro



AMD RX 7900 XTX

AMD Ryzen 7 5800X 8-Core Processor, 3793 MHz, 64GB RAM, Microsoft Windows 11 Pro



- Additional cache size info in the Device Configuration pane
- More Vector registers per SIMD
- Improved Wavefront
 occupancy for some
 workloads, including ray
 tracing events that use
 Indirect shader pipelines





RADEON GPU PROFILER 1.14 – RDNA 3 SUPPORT

mma_instructio Help	n_hip.rgp - HIP - Instruction tracing:	Full frame - Radeon GPU Pro	filer - V1.15.0.1							-	- 🗆	
	START OVERVIEW	EVENTS									SETTI	NC
avefront occ	supancy Event timing	Pipeline state	Instruction timing									
✓ Cijk_Ailk_E	3jlk_HHS_BH_MT128x128x32_MI1	6x16x16x1_SN_K1 🗸 🗸 Ev	vent 5				Wavefront					
Go to line	v_wmma_f32_16x16x	16_F16 🔇 1 of 112	V Clks normalized b	y hit counts	✓ Wavefront Latencies	: selection total	Latencies Histogram	473037 dk	Wavef	ronts: 22	949476 dk	k
				Hit count	Instruction cost (%)	Latency		Wavefront	statistics			Ē
542	ds load u16 d16 hi v19	1, v239 offset:2496		2794	0.18	10 clk		Timeline				
543	ds_load_u16_v192, v239	offset:2752		2794	0.16	9 clk						
544	ds_load_u16_d16_hi_v19	2, v239 offset:3008		2794	0.18	10 clk						
545	ds_load_u16 v193, v239			2794	0.15	8 clk						
546	ds_load_u16_d16_hi v19			2794	0.14	8 clk		Branches tot	al 3,014			
547	ds_load_u16 v194, v239			2794	0.11	6 clk						
548	ds_load_u16_d16_hi v19			2794	0.63	36 clk		Branches tak	en 94.16%			
549	buffer_load_b128 v[206	:209], V197, s[8:11], (Joffen	2794	0.02	l clk						
550 551	s_waitcnt lgkmcnt(0) v wmma f32 16x16x16 f1	6	-1120-1271	2794 2794	1.94 2.81	112 c1k 163 c1k		Instruction	n Hit count			
552	buffer load b128 v[210			2794	0.35	20 clk						
553	v wmma f32 16x16x16 f1			2794	1.91	110 clk		VALU	12,056			
554	buffer load b128 v[214			2794	0.32	18 clk		SALU	61,908			
555	v wmma f32 16x16x16 f1			2794	1.76	102 clk		JALO	01,500			
556	buffer load b128 v[218	:221], v200, s[8:11], () offen	2794	0.40	23 clk		VMEM	23,232			
557	v_wmma_f32_16x16x16_f1	6 v[24:31], v[163:170]	v[154:161], v[24:31	2794	1.72	99 clk						
558	buffer_load_b128 v[222	:225], v201, s[12:15],	0 offen	2794	0.33	18 clk		SMEM	66			
559	v_wmma_f32_16x16x16_f1				1.69	97 clk		LDS	743,424			
560	<pre>buffer_load_b128 v[226</pre>			2794	0.38	22 clk			,			
561	v_wmma_f32_16x16x16_f1				1.77	102 clk		IMMEDIATE	36,520			
562 563	buffer_load_b128 v[230			2794	0.43	25 clk		EXPORT	0			
563	v_wmma_f32_16x16x16_f1 buffer load b128 v[234			2794	1.84	106 c1k 29 c1k		EXPORT	U			
565	v wmma f32 16x16x16 fi				0.14	29 CIK		MISC	22			
566	s cmp eq u32 s5, s7	· · [00:00]; · ([::1:10]	·[104.101]/ •[00.03	2794	0.02	l clk						
567	s cselect b32 s66, s60	, s64		2794	0.02	l clk		RAYTRACE	0			
568	s cselect b32 s67, s61			2794	0.14	8 clk		WMMA	90.112			
569	v wmma f32 16x16x16 f1		v[130:137], v[64:71]	2794	0.14	8 clk			50,112			
570	s_add_u32_s8, s8, s66			2794	0.02	l clk		Total	975,986			
571	s_addc_u32 s9, s9, s67			2794	0.02	l clk						
572	s sub u32 s56, s56, s6	6		2794	0.14	8 clk						

- Additional cache size info in the Device Configuration pane
- More Vector registers on RDNA 3
- Improved Wavefront occupancy for some workloads, including ray tracing events that use Indirect shader pipelines
- WMMA instructions





- RGP can identify parts of the frame which perform inline raytracing
 - Pipelines
 - Events







- RGP can identify parts of the frame which perform inline raytracing ٠
 - Pipelines •

Frame summary	Pipeline sum	imary															
Barriers	All detected p	ipeline configurations															
Context rolls																	
Most expensive events	VS HS D	S GS PS CS RT 158 pi	pelines		VS HS DS GS	PS CS RT	66 pipelir	es									
Render/depth targets	VS HS D	S GS PS CS RT 1 pipe	line														
Pipelines																	
Device configuration	Pipelines																
	Bucket ID	Hash (API PSO + API shader) Duration	Event Count	Avg event duration	Occupancy	VGPRs	SGPRs	Scratch mem	Wave mode	Stag	25					
	> 214	0xCC879BC2012076A7	672.827 μs [4.05%]	1 [0.05%]	672.827 μs	16 - 16	68 - 68	42 - 42	No		VS	HS	DS	GS	PS	CS	RT
	> 86	0x53BE122CEF6FB004	231.513 µs [1.39%]	47 [2.52%]	18.121 µs	16 - 16	44 - 60	80 - 86	No		VS	HS	DS	GS	PS	CS	RT
	> 219	0x0103BAAFDFED89A9	224.824 µs [1.35%]	1 [0.05%]	224.824 µs	16 - 16	49 - 49	80 - 80	No		VS	HS	DS	GS	PS	CS	RT
	> 88	0x241BAF7E975C2995	220.115 µs [1.33%]	107 [5.74%]	20.707 µs	16 - 16	41 - 44	70 - 80	No		VS	HS	DS	GS	PS	CS	RT
	> 87	0xD86383187F807978	196.500 µs [1.18%]	42 [2.25%]	19.496 µs	16 - 16	38 - 44	66 - 80	No		VS	HS	DS	GS	PS	CS	RT
	> 89	0xF8570C5FAA57EB25	188.968 µs [1.14%]	46 [2.47%]	39.009 μs	16 - 16	44 - 54	80 - 102	No		VS	HS	DS	GS	PS	CS	RT
	> 216	0xA09B0F2620310864	183.828 µs [1.11%]	1 [0.05%]	183.828 µs	16 - 16	57 - 57	92 - 92	No		VS	HS	DS	GS	PS	CS	RT
	> 215	0xA23F9C83ED74A372	172.240 µs [1.04%]	1 [0.05%]	172.240 µs	16 - 16	68 - 68	42 - 42	No		VS	HS	DS	GS	PS	CS	RT







- RGP can identify parts of the frame which perform inline raytracing
 - Pipelines
 - Events

Frame summary													
Barriers						The most expensive 5% of events take 39% of all the	e time in the fram	e you a	re look	king at			
Context rolls										Ť			
Most expensive events													
Render/depth targets													
Pipelines													
Device configuration													
			20-25%		45-50%	70-75%			95	5-1009	%		
	-		20-25%		45-50%	70-75%		+	95	5-1004	%		
	 The region you	selected is app		. Your selection		70-75%		-	95	5-1009	% 		
	The region you Queue	selected is app		. Your selection			API	shader			% -		
			roximately 53% of the frame		is 100% graphics eve	nts and 0% asynchronous compute events.		shader HS	stage		-1	CS	RT
	Queue	Event ID	roximately 53% of the frame Event	Duration	is 100% graphics eve Work duration	nts and 0% asynchronous compute events.	VS	HS	r stage DS	es	-1	CS CS	
	Queue Č	Event ID	roximately 53% of the frame Event Dispatch(8160, 1, 1)	Duration 672.827 μs	is 100% graphics eve Work duration 672.827 μs	ents and 0% asynchronous compute events. User Event	VS	HS	r stage DS	es GS	PS	CS	RT RT
	Queue Direct queue Direct queue	Event ID 1724 147	roximately 53% of the frame Event Dispatch(8160, 1, 1) CmdDispatchBuildBVH()	Duration 672.827 μs 406.081 μs	is 100% graphics eve Work duration 672.827 μs 406.081 μs	ents and 0% asynchronous compute events. User Event	VS e=Top) VS	HS HS	DS DS	es GS GS	PS PS	CS CS	RT RT
	Queue Direct queue Direct queue Direct queue	Event ID 1724 147 1731	roximately 53% of the frame Event Dispatch(8160, 1, 1) CmdDispatchBuildBVH() Dispatch(60, 34, 1)	Duration 672.827 μs 406.081 μs 224.824 μs	is 100% graphics eve Work duration 672.827 μs 406.081 μs 224.824 μs	ents and 0% asynchronous compute events. User Event	VS e=Top) VS VS	HS HS HS	DS DS DS	es GS GS GS	PS PS PS	CS CS	RT





GPU Profile

- RGP can identify parts of the frame which perform inline raytracing
 - Pipelines
 - Events

avefront occu	upancy Event timing	Pipeline state	Instruction timing						
V API PSO 0x	CC879BC2012076A7 V Event 1	724 VS	HS DS GS PS	CS RT				Wavefront Latencies	ويطلطه
Go to line	image_bvh	3 1 of 2	V Clks normalized by wav	efronts 🛛 🗸 Wavefront La	atencies: sele	ction total		Histogram 2164 dk	
	,				Hit coun	t Instruction cost (%)	Latency		
450	s delay alu instid0(VALU D	EP 1) instskip(NEX	T) instid1 (VALU DEP 4)		9943	0.00		Instau	tion Hit co
451	v add co u32 v35, vcc lo,	v44, v21			9943	0.11	247 clk	Instruc	
452	v add co ci u32 e64 v23, v	cc_lo, v22, 0, vcc_l	0		9943	0.15	320 clk	VALU	857,597
453	v and b32 e32 v22, 0xfffff	fc0, v24			9943	0.15	314 clk		
454	v rcp f32 e32 v21, v38				9943	0.22	470 clk	SALU	353,744
455	s delay alu instid0(VALU D	EP 2) instskip(NEX	T) instid1 (VALU DEP 2)		9943	0.00		10.001	4 600
456	v_or_b32_e32_v36, v23, v2	-			9943	0.15	316 clk	VMEM	1,608
457	v_add_co_u32 v24, vcc_lo,	v22, v5			9943	0.15	329 clk	SMEM	1,902
458	v_rcp_f32_e32_v22, v39				9943	0.12	261 clk	SHEH	1,502
459	v add co ci u32 e64 v25, v	cc_lo, v6, 0, vcc_lo			9943	0.12	251 clk	LDS	13,301
460	v_cmp_le_u32_e32 vcc_lo, 4	, v 33			9943	0.21	455 clk		
461	v rcp f32 e32 v23, v40				9943	1.37	2,976 clk	IMMEDI.	ATE 192,179
462	image_bvh64_intersect_ray	v[33:36], [v[35:36],	v28, v[41:43], v[38:40],	v[21:23]], s[16:19]	9943	0.08	164 clk	EVE OF	
463	s_and_saveexec_b32_s3, vcc	10			9943	0.54	1,169 clk	EXPORT	0
464	s_cbranch_execz _L13				9943	0.27	585 clk	MISC	170
465 BBF0_10:								Wise	170
466	<pre>s_and_saveexec_b32 s2, s2</pre>				9405	0.49	1,056 clk	RAYTRA	E 13,301
467	s_cbranch_execz_L14				9405	0.77	1,670 clk		
468 BBF0_11:								WMMA	0
469	s_clause 0x4				5648	0.05	107 clk		
470	global load b128 v[37:40],	v[24:25], off			5648	0.04	92 clk	Total	1,531,3





GPU Profile

- RGP can identify parts of the frame which perform inline raytracing
 - Pipelines
 - Events
 - Wavefronts







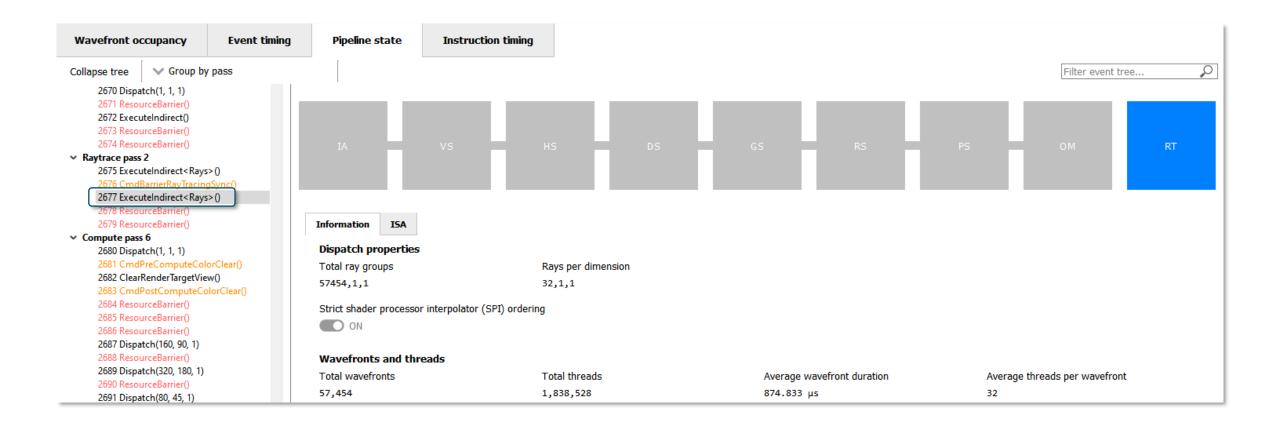
- RGP can identify parts of the frame which perform inline raytracing
 - Pipelines
 - Events
 - Wavefronts







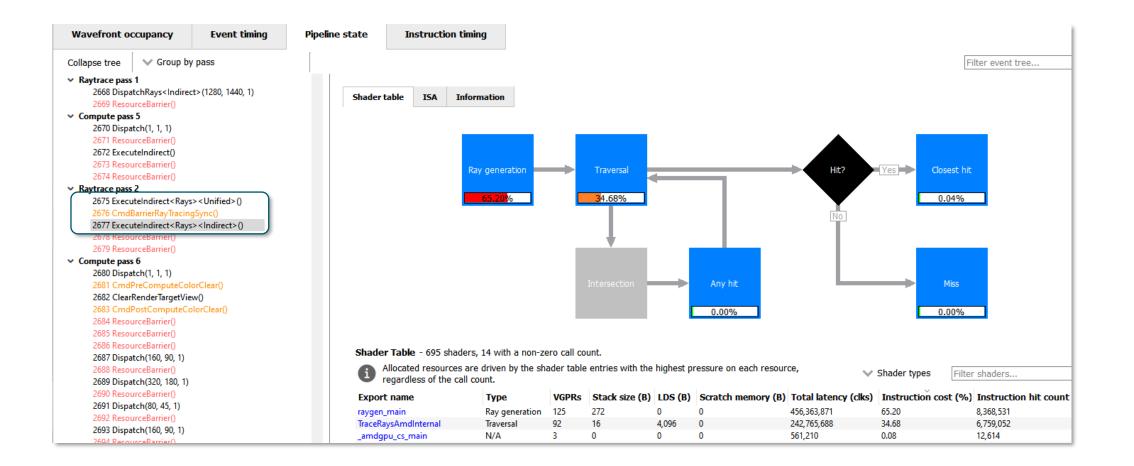
• Support for ExecuteIndirect calls that perform ray tracing







• Support for ExecuteIndirect calls that perform ray tracing







GPU Profiler

RADEON GPU PROFILER 1.14 - HIP/OPENCL SUPPORT

Phip_transform-202	230212-13	4523.rgp - HIP - Ins	truction tracir	ıg: Full fra	me - Radeon GPU	Profiler -
ST	ART	OVERVIEW	EVENTS			
Wavefront occu	pancy	Event timing	Pipeline	state	Instruction	timing
Collapse tree	🗸 Col	or by event				
29 MatrixMul 30 CmdBarrier 31 hipMemset 32 CmdBarrier 33 hipMemset 34 CmdBarrier 35 Transform 36 CmdBarrier 37 SimpleCon 38 CmdBarrier 39 ComputeN 40 CmdBarrier 41 ReorderDat 42 CmdBarrier 43 hipMemset 44 CmdBarrier 45 CollapseNo	c) c) t() c) volution c) lesh c) ta c) ta c) t() c) c)	26,700,	000.000 μs	0.181 μs 2.647 μs 0.181 μ 2.557 μ 2.609 μ 1.222 μ 0.1 2.99 3.0 2.9 94. 94. 3.0 0.1	s s	26,9
46 CmdBarrier 47 hipMemse 48 CmdBarrier 49 InitializeDa	t() r()			2.6	181 µs 543 µs 0.181 µs 8.620 µs	

		1		1
Go to line	Search 🔎 N	lo res 3	Clks normalized by h	nit counts V Wavefront Latencies: selection tota
	1	5 Hit ₇	Instruction cost (%)	Latency
85	s branch L1	51 9	0.01	ll clk
86 BBF0_1:		11		· · · · · · · · · · · · · · · · · · ·
37	s nop 0	0	p.00	
88 L3 :	_	13		
89	<pre>s_or_b32 exec_lo, exec_lo, s0</pre>	65 15	0.11	l clk
90	s_add_i32 s0, s14, -2	65 17	0.11	l clk
91	s_cmp_lt_u32 s14, 3	65 19	0.11	l clk
92	s mov b32 s14, s0	65 21	p.11	l clk
93	<pre>s_waitcnt vmcnt(0) lgkmcnt(0)</pre>	65 23	8.11	77 clk
94	s_waitcnt_vscnt_null, 0x0	65 25	3.76	35 clk
95	s_barrier	65 27	9.01	85 clk
96	<pre>s_waitcnt vmcnt(0) lgkmcnt(0)</pre>	65 29	p.11	l clk
97	<pre>s_waitcnt_vscnt null, 0x0</pre>	65	p.18	l clk
98	buffer_gl0_inv	65278	0.12	l clk
99	s_cbranch_scc1 _L0	65278	1.74	16 clk
00 _L1:				
01	v_cmp_gt_u32_e32 vcc_lo, s14, v0	65278	1.77	16 clk
02	<pre>s_and_saveexec_b32 s0, vcc_lo</pre>	65278	0.97	9 clk
03	s_cbranch_execz _L2	65278	2.81	26 clk
04 BBF0_2:				
05	<pre>ds_read_b128 v[11:14], v1 offset:16</pre>	48830	0.75	9 clk
06	ds_read_b128 v[15:18], v1	48830	0.08	l clk
07	<pre>s_waitcnt lgkmcnt(1)</pre>	48830	12.97	164 clk
08	v mul f32 e32 v11, v6, v11	48830	0.08	l clk





GPU Profiler

RADEON GPU PROFILER - UPCOMING WORK - NEW ISA VIEW

• Motivation – differences in current ISA views

sle_profile.rgp - D3D12 - Instruction tracing: Full f	me - Radeon GPU Profiler - V1.14.0.3		O sample_profile.rgp - D3D12 - Instruction tracing: Full frame - Radeon GPU Profiler - V1.14.0.3			-
START OVERVIEW	EVENTS		File Help START OVERVIEW EVENTS			
STARI OVERVIEW	EVENIS	SETTIN	START OVERVIEW EVENTS			
ont occupancy Event timing	Pipeline state Instruction timing		Wavefront occupancy Event timing Pipeline state Instruction timing			
tree 🛛 🗸 Group by user events		Filter event tree	✓ API PSO 0x43E702185888408CC ✓ Event 5 VS HS DS GS PS CS		Wavefror Latencie	
Resource() rceBarrier() arrierRidSunc()			Go to line D No results V Clks normalized by hit counts	✓ Wavefront Latencies: selection total	Histogram	
BarrierBlitSync() urceBarrier()				Hit count Instruction cost (%)	Latency	Geometry shader
enderTargetView()	IA VS HS DS GS	RS PS OM CS	1 s_bfe_u32 s0, s3, 0x80000	3 0.08	3 clk	5 DrawInstanced(16384, 1, 0, 0)
rticles			2 s bfm b64 exec, s0, 0 3 s cmp eq u32 s0, 64	3 0.03	1 clk	Draw particles
nstanced(16384, 1, 0, 0)			4 s capiect b4 exec1, exec	3 0.03	l clk l clk	brain particles
lart			5 s getpc b64 s[6:7]	3 0.13	5 clk	Identifiers
Step 0			6 s_inst_prefetch 0x3	3 0.03	1 clk	API shader hash
purceBarrier()	Information ISA		7 s_cmp_eq_u32_s0, 0	3 0.23	9 clk	0x5905EA7F390FFC7681D23588F7A6D8F1
atch(64, 1, 1)	amornaooni asv		8 s_cbranch_soc1_L0 9 BBP0 0:	3 0.08	3 clk	API PSO hash
ourceBarrier()	Shader ISA 📋 Search 🔎 No results		10 s_mov_b32 s0, s20	3 0.03	1 clk	0x43E70218588408CC
ResolveQueryTimestampStall() ResolveQueryCopy()	// API shader hash: 0x858B88B360FD480E4BEA8AF99C466046		11 s_mov_b32 s1, s7	3 0.03	1 clk	Driver internal pipeline hash
and a start of the	// API shader hash: 0x838B08B3607048054B2A8A759C466046 // API PSD hash: 0x438720185B8408CC	î.	12 v_mbcnt_lo_u32_b32 v7, -1, 0	3 0.20	8 clk	0xDB2B590CBE6A24A243E702185BB408CC
	// Driver internal pipeline hash: 0x805590CBE6A24A243E702185BB408CC		13 s_load_dwordx4 s[24:27], s[0:1], null	3 0.03	1 clk	
	11		14 s_mov_b32 s0, s12 15 s_load_dwords4 s[28:31], s[0:1], mull	3 0.23	9 clk 1 clk	and the standard second s
	// Vector registers: 32 (32 allocated)		15 s bfe u32 s0, s3, 0x40018	3 0.03	1 clk	Wavefront statistics
	// Scalar registers: 32 (128 allocated)		17 s mulk 132 s0, 0x900	3 0.03	1 clk	Timeline
	andgpu gs main:		18 v add nc u32 e32 v6, s13, v5	3 0.03	1 clk	
	s_bfe_u32_s0, s3, 0x80000	// 000000000000 9380FF03 00080000	19 s.waitcnt lgkmcnt(0) 20 thuffer load format xymw v[8:11], v6, s[24:27], 0 format:[BUF FNT 32 32 32 32 37	3 10.55 FLDAT] idxen 3 0.05	415 clk	
	s_bfm_b64 exec, s0, 0	// 00000000008: 92F28000 // 0000000000C: B706C000	 20 tbuffer load format xyzw v[8:11], v6, s[24:27], 0 format:[BUF FMT 32 32 32 32 FL 21 tbuffer load format x v6, v5, s[28:31], 0 format:[BUF FMT 32 FLDAT] idxen offse 	ti28 3 0.05	2 clk 3 clk	Branches total 6
	s cmp eq u32 s0, 64 s cselect b64 exec, -1, exec	// 00000000000: BF06C000 // 0000000001: 85F27EC1	22 thuffer load format xyz v[12:14], v5, s[28:31], 0 format:[BUF_PMT_32_32_32_FLOA 22 thuffer_load_format xyz v[12:14], v5, s[28:31], 0 format:[BUF_PMT_32_32_32_FLOA		12 clk	
	s getpc b64 s16:7]	// 0000000014: B861280	23 v abont hi u32 b32 v5, -1, v7	3 0.13	5 clk	Branches taken 0.00%
	s_inst prefetch 0x3	// 00000000018: BFA00003	24 v mad u32 u24 v5, v5, 36, s0	3 0.03	1 clk	
	s_cmp_eq_u32 s0, 0	// 0000000001C: BF068000	25 s_waitent vment(2)	3 29.69	1,168 clk	Instruction Hit count
	s_cbranch_sccl_L0 BBF0_0:	// 0000000020: BF950033	26 v add f32 e32 v7, -1.0, v8 27 v add f32 e32 v8, 0xbdccccd, v9	3 0.05	2 clk	VALU 387
	s mov b32 s0, s20	// 0000000024: BE800314	27 V add_f32_e32 V#, 0xbdcccccd, V9 28 V add_f32_e32 V#, 0xbdcccccd, V10	3 0.05	2 clk 1 clk	
	s mov b32 s1, s7	// 0000000028: BE810307	29 s waitent vment(1)	3 7.13	280 clk	SALU 72
	v mbcnt_lo_u32_b32_v7, -1, 0	// 0000000022: D7650007 000100C1	30 v mul f32 e32 v6, 0x3de38e39, v6	3 0.14	5 clk	VMEM 9
	<pre>s load dwordx4 s[24:27], s[0:1], null</pre>	// 0000000034: E4080600 FA000000	31 v madak_f32 v10, v6, v9, 0x3dcccccd	3 0.05	2 clk	
	<pre>s nov b32 s0, s12 s load dwords4 s[28:31], s[0:1], null</pre>	// 0000000003C: BK80030C // 00000000040: F4080700 FA000000	32 v_add_f32_e32 v9, -1.0, v11	3 0.05	2 clk	SMEM 18
	s ble u3 s0, s3, 0340018	// 00000000048: 9380FF03 00040018	33 v_mad_f32 v7, v6, v7, 1.0 34 v_madak_f32 v8, v6, v8, 0x3dcccccd	3 0.05	2 clk 2 clk	LDS 84
	s mulk i32 s0, 0x900	// 00000000050: B8000900	34 V madak_132 v8, v6, v9, 0x3acccccd 35 v mad f32 v11, v6, v9, 1.0	3 0.03	2 clk	
	v_add_nc_u32_e32_v6, s13, v5	// 00000000054: 4A0C0A0D	36 s waitent vment(0)	3 0.42	16 clk	IMMEDIATE 36
	<pre>s waitent lghment(0) thuffer load format xyzw v[8:11], v6, s[24:27], 0 format:[BUF_FMT_32_32_32_32_FLOAT] idxen</pre>	// 00000000058: BF8CC07F // 0000000055C: EAGB2000 80060806	37 ds_write2_b32_v5, v12, v13 offset1:1	3 0.05	2 clk	EXPORT 0
	thuffer load format x v6, v5, s[28:31], 0 format:[BUF FMT 32 FLDAT] idxen offset:28	// 0000000064: E880201C 80070605	38 ds write b32 v5, v14 offset:8 36 ds write2 b32 v5, v2, v8 offset0:4 offset1:5	3 0.05	2 clk	
	tbuffer_load_format_xyz v[12:14], v5, s[28:31], 0 format:[BUF_FMT_32_32_32_FLOAT] idxen	// 0000000006C: EA522000 80070C05	39 ds write2 b32 v5, v7, v8 offset0:4 offset1:5 40 ds write2 b32 v5, v10, v11 offset0:6 offset1:7	3 0.05	2 clk 1 clk	MISC 18
	v mbont hi ul2 b32 v5, -1, v7	// 0000000074: D7660005 00020EC1	41 L0:	5 0103	1 016	Total 630
	v mad u32 u24 v5, v5, 36, s0 s waitent vment(2)	// 0000000007C: D5430005 00014905 // 00000000084: BF8C3F72	42 s_vaitcnt vmcnt(0) expcnt(0) lgkmcnt(0)	3 1.47	57 clk	
	v add f32 e32 v7, -1.0, v0	// 00000000088: 06021023	43 s barrier	3 0.03	1 clk	
	v add f32 e32 v8, 0xbdcccccd, v9	// 0000000008C: 061012FF BDCCCCCD	44 s_bfe_u32 s4, s3, 0x80008	3 0.05	2 clk	Hardware utilization
	v_add_f32_e32_v9, 0xbdcccccd, v10	// 00000000094: 061214FF BDCCCCCD	45 s_bfm_b64 exec, s4, 0 46 s_cmp_eq_u32 s4, 64	3 0.03 3 0.03	l clk l clk	
	s vaitent vmcnt(1) v mul f32 e32 v6. 0x3de38e39. v6	// 0000000009C: BF8C3F71 // 000000000000: 100C0CFF 3DE38E39	47 s_cselect_b64 exec, -1, exec	3 0.03	1 clk	
	v madak [32 v10, v6, v9, 0x3dcccccd	// 000000000A8: 42141306 3DCCCCCD	48 s_inst_prefetch 0x3	3 0.03	1 clk	
	v add f32 e32 v9, -1.0, v11	// 00000000000: 061216F3	49 s_bfe_u32 s3, s3, 0x80010	3 0.05	2 clk	
	v_mad_f32 v7, v6, v7, 1.0	// 000000000B4: D5410007 03CADF06	50 s mov b32 m0, s3 51 s cmp eq u32 s4, 0	3 0.03	1 clk	4.704
	v_madak_f32 v8, v6, v8, 0x3dcccccd v_mad_f32 v11, v6, v9, 1.0	// 000000000BC: 42101106 3DCCCCCD // 000000000C4: D541000B 03CA1306	51 s_cmp_eq_u32 s4, 0 52 s_cbranch_scc1_L1	3 0.23	9 clk 3 clk	6.7% 0.6% 0.6% 0.2% 2.9%
	s waitent vmcnt(0)	// 000000000CC: BF9C3F70	53 BBF0 1:		5 04A	VALU SALU VMEM SMEM LDS
	ds write2 b32 v5, v12, v13 offset1:1	// 00000000000: D8380100 000D0C05	54 v mbcnt_lo_u32_b32 v1, -1, 0	3 0.15	6 clk	Shader statistics
	da vrite b32 v5, v14 offset:8	// 000000000B8: D#340008 00000Z05	55 s_add_u32 s0, s21, s2	3 0.03	1 clk	
	<pre>ds write2 b32 v5, v7, v0 offset0:4 offset1:5 ds write2 b32 v5, v10, v11 offset0:6 offset1:7</pre>	// 00000000028: D8380504 00080705 // 00000000028: D8380706 00080A05	56 v mov b32_e32 v3, 0 57 v mbont hi u32_b32 v1, -1, v1	3 0.05	2 clk	Shader duration: 0.039 ms
	LO:		57 v_mbcnt_hi_u32_b32 v1, -1, v1 58 v ishirev b32 sdwa v0, 2, v0 dst sel:DWDRD dst unused:UNUSED FAD src0 sel:DWDRD		2 clk 4 clk	Wavefronts: 3 out of 256 analyzed
	<pre>s_waitcnt vmcnt(0) expcnt(0) lgkmcnt(0)</pre>	// 00000000000: BF8C0000	59 s pack 11 b32 b16 s1, s11, 16	3 0.03	l clk	Theoretical occupancy: 16 / 20 total wavefronts p
	s barrier	// 000000000F4: BF8A0000	60 v mul u32 u24 e32 v1, Oxc4, v1	3 0.05	2 clk	Vector registers: 32 (32 allocated)
	s bfe u32 s4, s3, 0x80008 s bfn b64 exec, s4, 0	// 000000000F8: 9384FF03 00080008 // 00000000100: 92FE004	61 s mov b32 s4, s10	3 0.03	1 clk	Scalar registers: 32 (128 allocated)
	s_ptm_pose exec, ss, u s cmp eq u32 s4, 64	// 00000000014: BP06C004	62 s_mov_b32 s5, s1	3 0.03	1 clk	
	s cselect b64 exec, -1, exec	// 00000000108: 85FE7EC1	63 s_movk_i32 s6, 0x1000 64 s_mov_b32 s7, 0x2104bfac	3 0.03	1 clk 1 clk	Call targets
	s_inst_prefetch 0x3	// 0000000010C: BFA00003	65 v add nc u32_e32 v25, 64, v1	3 0.21	l cik B cik	Please select a jump instruction (s_setpc, s_swap
	s bfe u32 s3, s3, 0x80010 s mov b32 m0, s3	// 00000000110: 9383FF03 00080010 // 00000000118: EEFC0303	66 s buffer load dwordx4 s[8:11], s[4:7], 0x50	3 0.03	1 clk	
	s_mov_b32 m0, s3 s_cmp_eq_u32 s4, 0	// 00000000118: BEFC0303 // 0000000011C: BF068004	67 s_buffer_load_dwordx4 s[12:15], s[4:7], 0x40	3 0.03	1 clk	
	s chranch scol Ll	// 00000000120: BF8500E6	68 s_buffer_load_dwordx8 s[16:23], s[4:7], null	3 0.03	l clk	
	BBTO 1:		69 s_buffer_load_dwordx8 s[24:31], s[4:7], 0x20 70 v add nc u32 e32 v2, s0, v1	3 0.04 3 0.05	1 clk	
	- v mbcmt lo u32 b32 v1 -1 0				2 clk	





AMDJ RADEON Raytracing Analyzer



Detailed video of RRA

"Introduction to Raytracing with the Radeon Developer Tools Suite"

gpuopen.com/videos

https://www.youtube.com/watch?v=i4Evh4SJtUg



AMD PUBLIC | GDC 23 | OPTIMIZING GAME PERFORMANCE WITH THE RADEON DEVELOPER TOOL SUITE | MARCH 2023

RADEON DRIVER - NEW RAYTRACING FEATURES

Instance Rebraiding and Triangle Splitting

- Improved spatial coherence
- Overall traversal speedup
- May occupy larger memory
- Might take longer to build
- Automatically activated by the driver
- No user input required!



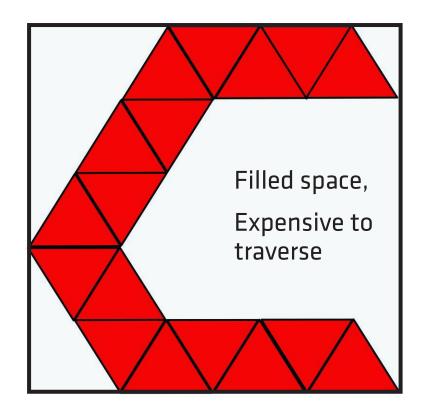




RADEON RAYTRACING ANALYZER 1.1 - UPDATE

Instance Rebraiding

- Instances are automatically rebraided based on transform and BLAS complexity by the driver
- The immediate child nodes of BLAS root will be encoded into the TLAS when the instance is rebraided
- Traversal loop may perform less iterations as a result of reduced surface area



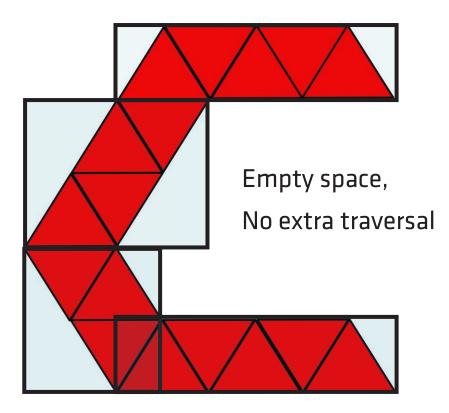




RADEON RAYTRACING ANALYZER 1.1 - UPDATE

Instance Rebraiding

- Instances are automatically rebraided based on transform and BLAS complexity by the driver
- The immediate child nodes of BLAS root will be encoded into the TLAS when the instance is rebraided
- Traversal loop may perform less iterations as a result of reduced surface area





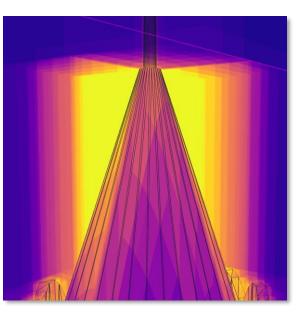




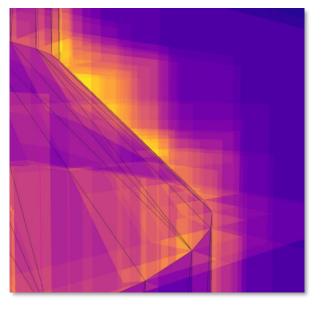
RADEON RAYTRACING ANALYZER 1.1 UPDATE

Triangle Splitting

- Triangle nodes are automatically split based on size and orientation by the driver to provide better early termination in raytriangle tests
- If a triangle node is split, the parent
 bounding boxes will overlap with parts of the triangle but still point to the same geometry and primitive index



Without triangle splitting, Compounding traversal cost



With triangle splitting Reduced traversal cost

(Heatmap - yellow is more expensive to traverse)





AMDJ RADEON Memory Visualizer



Detailed video of RMV

"Curing Amnesia and Other GPU Maladies with AMD Developer Tools"

gpuopen.com/videos

http://www.youtube.com/watch?v=2tmLQVn36P8?t=24m48s



AMD PUBLIC | GDC 23 | OPTIMIZING GAME PERFORMANCE WITH THE RADEON DEVELOPER TOOL SUITE | MARCH 2023

RADEON MEMORY VISUALIZER 1.4.1 - UPDATE

Introduced full capture and visualization on Linux

Activities	🕅 RadeonMemoryVisu	sualizer May 20 17:01	🚣 🌒 🕛
_			
		/home/RDTS/sample trace.rmv - Radeon Memory Visualizer	- • ×
	File Help		
A	4) ST	TART TIMELINE SNAPSHOT COMPARE	
?	Snapshot generation Device configuration	Snapshot timeline Right-click on the timeline below to generate a snapshot at that moment in time.	
		Resource usage size Compare snapshots	
•			32.50s
·M		Texture: 2.72 GB	
. D		Render target: 640.63 MB Deph sternit Juhrer 245.63 MB Index buffer: 125.63 MB Vertex buffer: 125.00 MB	
0		Other: 538.79 KiB	
_		Depth stencil buffer Render target Command buffer Render target Vertex buffer Index buffer Ray tracing UAV Shader pipeline Command buffer Heap Descriptors Multi-use buffer Event Unbound Internal Selection: O	i.63s 21.82s
0		Search	
		Snapshot name Timestamp * Virtual allocations Resources Total virtual memory Bound virtual memory Unbound virtual memory Committed local memory Committed invisible memory Committed host i	iemory
0		Snapshot 0 05.01s	
-		Snapshot 1 12.90s	
		Snapshot 2 31.22s	
		Snapshot count: 3	

* Note: Raytracing support dependent on AMD driver version

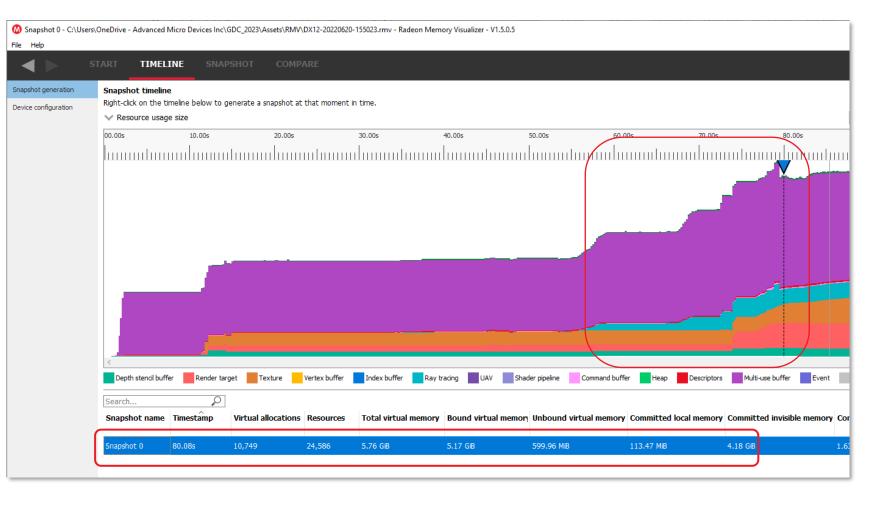






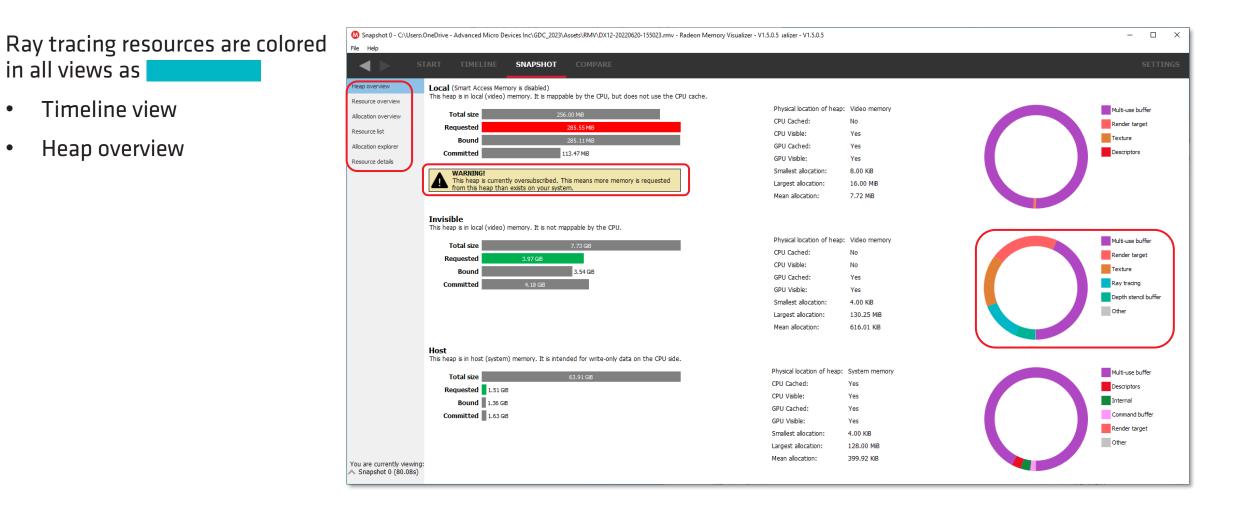
Ray tracing resources are colored in all views as

• Timeline view











•

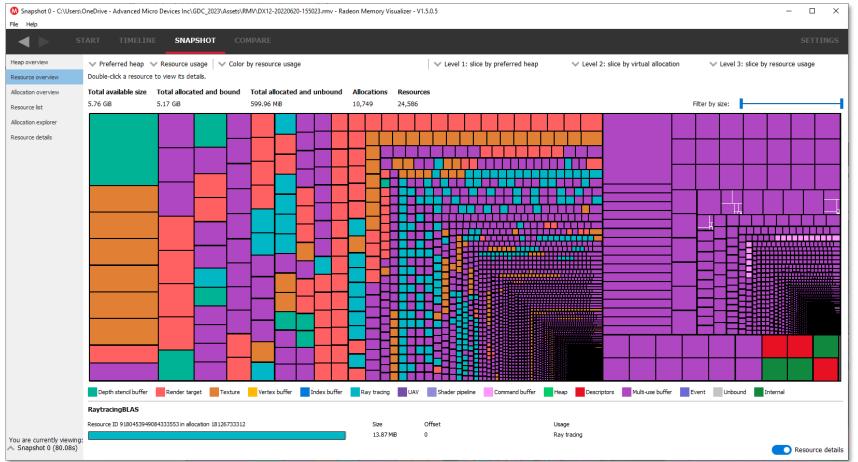
۲



Memory Visualizer

Ray tracing resources are colored in all views as

- Timeline view
- Heap overview
- Resource overview



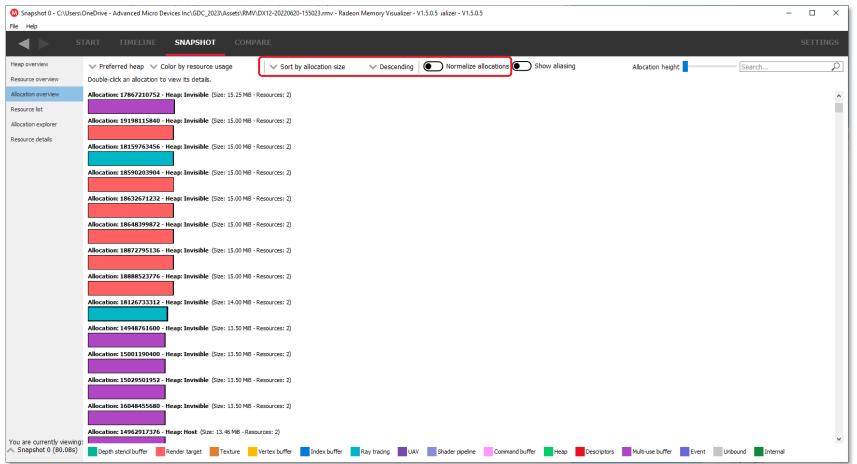




Memory Visualizer

Ray tracing resources are colored in all views as

- Timeline view
- Heap overview
- Resource overview
- Allocation overview









Ray tracing resources are colored in all views as

- Timeline view
- Heap overview
- Resource overview
- Allocation overview

	pselley\OneDrive - Advanced Micro Devices Inc\GDC_2023\Assets\RMV\DEMO_DX12-20220620-155023.rmv - Radeon Memory Visualizer - V1.5.0.5 – 🗆 🗙
File Help	
ST	ART TIMELINE SNAPSHOT COMPARE SETTINGS
Heap overview	V Preferred heap V Color by resource usage V Sort by fragmentation score V Descending Normalize allocations 💽 Show aliasing Allocation height Search
Resource overview	Double-click an allocation to view its details.
Allocation overview	Allocation: 4315938816 - Heap: Invisible (Size: 1.00 MB - Resources: 167)
Resource list	
Allocation explorer	Allocation: 4340056064 - Heap: Invisible (Size: 2.00 MB - Resources: 150)
Resource details	
	Allocation: 4295229440 - Heap: Local (Size: 256.00 KB - Resources: 125)
	Allocation: 12941524992 - Heap: Invisible (Sze: 256.00 KB - Resources: 64)
	Allocation: 12941787136 - Heap: Invisible (Size: 256.00 KB - Resources: 64)
	Allocation: 12943097856 - Heap: Invisible (Size: 256.00 KB - Resources: 64)
	Allocation: 12943360000 - Heap: Invisible (Sze: 255.00 KB - Resources: 64)
	Allocation: 12945457152 - Heap: Invisible (Size: 256.00 KB - Resources: 64)
	Allocation: 12946505728 - Heap: Invisible (Size: 256.00 KB - Resources: 64)
	Allocation: 12947292160 - Heap: Invisible (Sze: 256.00 KB - Resources: 64)
	Allocation: 12947554304 - Heap: Invisible (Size: 256.00 KB - Resources: 64)
	Allocation: 12947816448 - Heap: Invisible (Size: 256.00 KB - Resources: 64)
	Allocation: 12948078592 - Heap: Invisible (Size: 256.00 KB - Resources: 64)
Maria and a state of the state	Allocation: 12948340736 - Heap: Invisible (Sze: 256.00 KB - Resources: 64)
You are currently viewing: Snapshot 0 (80.08s)	Depth stenci buffer Render target Texture Vertex buffer Index buffer Ray tracing UAV Shader pipeline Command buffer Heap Descriptors Multi-use buffer Event Unbound Internal





AMD RADE

Memoru Visualizer

Ray tracing resources are colored in all views as

- Timeline view
- Heap overview
- Resource overview
- Allocation overview
- Allocation explorer

		IELINE SN	APSHOT										
view	Search	ĥ	D									Filter by size:	
verview	Allocation	Allocation size	Bound	Unbound	Avg. resource size	Resource size std. dev.	Resource count	Preferred heap	Committed invisible	e Committed local	Committed host	t Unmapped	
verview							_						
t	12904038400		0 bytes	256.00 KiB	0 bytes	0 bytes	0	Invisible	0 bytes	0 bytes	0 bytes	256.00 KiB	
xplorer	12903776256		0 bytes	256.00 KiB	0 bytes	0 bytes	0	Invisible	0 bytes	0 bytes	0 bytes	256.00 KiB	
etails	12903514112		0 bytes	256.00 KiB	0 bytes	0 bytes	0	Invisible	0 bytes	0 bytes	0 bytes	256.00 KiB	
	12903251968		0 bytes	256.00 KiB	0 bytes	0 bytes	0	Invisible	0 bytes	0 bytes	0 bytes	256.00 KiB	
	12902989824	256.00 KiB	0 bytes	256.00 KiB	0 bytes	0 bytes	0	Invisible	0 bytes	0 bytes	0 bytes	256.00 KiB	
	12902727680	256.00 KiB	0 bytes	256.00 KiB	0 bytes	0 bytes	0	Invisible	0 bytes	0 bytes	0 bytes	256.00 KiB	
	12902465536	256.00 KiB	0 bytes	256.00 KiB	0 bytes	0 bytes	0	Invisible	0 bytes	0 bytes	0 bytes	256.00 KiB	
(4314234880	256.00 KiB	152.63 KiB	103.38 KiB	5.87 KiB	1,005 bytes	26	Invisible	256.00 KiB	0 bytes	0 bytes	0 bytes	
	4295229440			104.44 KiB		42 bytes	125	Local	0 bytes	0 bytes	0 bytes	256.00 КВ	
	4295229440			104.44 KiB Show aliasin		42 bytes	125		0 bytes	0 bytes	0 bytes	256.00 KB	
		source usage		Show aliasin	9	42 bytes			0 bytes				rnal
	✓ Color by re	esource usage		Show aliasin	9								rnal
	Color by re	esource usage	target T	Show aliasin	9	buffer Ray tracing	UAV Shader p	ipeline Comman		Descriptors Mu		vent Urbound Inter	rnal
	Color by re Depth stend Search Name	esource usage	target T Virtual ad	Show aliasin exture Ve dress Size	g ertex buffer Index Preferred h	buffer Ray tracing	UAV Shader p	ipeline Comman	d buffer Heap	Descriptors Mu		vent Urbound Inter	mal
· ·	Color by re Depth stend Search Name Resource 9165	esource usage	target T Virtual ad 3 0x1000400	Show aliasin exture Ve dress Size	g ertex buffer Index Preferred h L KIB Local	buffer Ray tracing	UAV Shader p	ipelne Comman	d buffer Heap	Descriptors Mu		vent Urbound Inter	mal
· ·	Color by re Depth stend Search Name Resource 9165 Resource 9115	buffer Render	target Transformer Virtual ad 3 0x1000400 0x1000440	Show aliasin exture Ve dress Size 100 14.8:	g ertex buffer Index Preferred h L KIB Local oytes Local	buffer Ray tracing eap Committed invisib 0 bytes	UAV Shader p le Committed loc 0 bytes	ipeline Comman	d buffer Heap bst Unmapped Usar 14.81 KiB Shac 768 bytes Shac	Descriptors Mu Je		vent Urbound Inter	mal
· ·	Color by re Depth stend Search Name Resource 9165 Resource 1115 Resource 1700	source usage	target T Virtual ad 0x1000400 0x10004400 0x1000440	Show aliasin exture Ve dress Size 100 14.8: 100 768 100 768 100 960	g ertex buffer Index Preferred h L KIB Local pytes Local	buffer Ray tracing eap Committed invisib 0 bytes 0 bytes	UAV Shader p le Committed loc 0 bytes 0 bytes	pelne Comman al Committed ho 0 bytes 0 bytes	d buffer Heap bt Unmapped Usar 14.81 KIB Shac 768 bytes Shac 960 bytes Shac	Descriptors Mu ge ler pipeline ler pipeline		vent Urbound Inter	mal





Ray tracing resources are colored in all views as

- Timeline view
- Heap overview
- Resource overview
- Allocation overview
- Allocation explorer
- Resource List

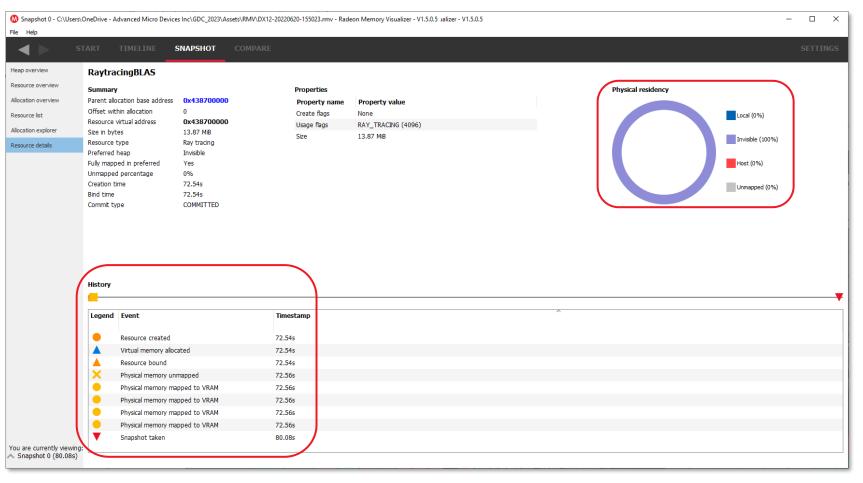
•	neDrive -	Advanced Micro Devices Inc	c\GDC_2023\Assets\F	RMV\DX12-2022	20620-155023.rmv - R	adeon Memory Visualize	er - V1.5.0.5 Jalizer - V	1.5.0.5			- 🗆 X
	ART	TIMELINE SNA	APSHOT CO	MPARE							SETTINGS
File Help											
Resource list Allocation explorer						Heaps (9,498) Textures (4,07 Shader pipelines Ray tracings (21 Render targets	5) s (656) L6)			Invisible	3.97 GiB out of 7.73 GiB
		erred heap 🛛 🗸 Resource		- T	Preferred heap	Committed invisible	Committed local	Committed host	Unmapped	Usage	Search
	Raytraci	ingBLAS	0x438700000	13.87 MiB	Invisible	13.87 MiB	0 bytes	0 bytes	0 bytes	Ray tracing	
	Raytraci	ingBLAS	0x439980000	617.25 KiB	Invisible	617.25 KiB	0 bytes	0 bytes	0 bytes	Ray tracing	
	Raytraci	ingBLAS	0x439780000	2.00 MiB	Invisible	2.00 MiB	0 bytes	0 bytes	0 bytes	Ray tracing	
	Raytraci	ingBLAS	0x43a0c0000	528.75 KiB	Invisible	528.75 KiB	0 bytes	0 bytes	0 bytes	Ray tracing	
	START TIMELINE SNAPSHOT COMPARE Virtual memory Bound virtual memory Unbound virtual memory Verterred heap< ◆ Resource usage Filter by stee:	2.00 MiB	Invisible	2.00 MiB	0 bytes	0 bytes	0 bytes	Ray tracing			
	Raytraci	ingBLAS	0x43a680000	14.83 MiB	Invisible	14.83 MiB	0 bytes	0 bytes	0 bytes	Ray tracing	
	Raytraci	ingBLAS	0x43cf80000	2.00 MiB	Invisible	2.00 MiB	0 bytes	0 bytes	0 bytes	Ray tracing	
	Raytraci	ingBLAS	0x4731c0000	1.79 MiB	Invisible	1.79 MiB	0 bytes	0 bytes	0 bytes	Ray tracing	
	Raytraci	ingBLAS	0x473640000	1.79 MiB	Invisible	1.79 MiB	0 bytes	0 bytes	0 bytes	Ray tracing	
	Raytraci	ingBLAS	0x473ac0000	1.79 MiB	Invisible	1.79 MiB	0 bytes	0 bytes	0 bytes	Ray tracing	
	Raytraci	ingBLAS	0x4743c0000	1.79 MiB	Invisible	1.79 MiB	0 bytes	0 bytes	0 bytes	Ray tracing	
	Raytraci	ingBLAS	0x4745c0000	1.79 MiB	Invisible	1.79 MiB	0 bytes	0 bytes	0 bytes	Ray tracing	
ou are currently viewing: Snapshot 0 (80.08s)	Total res	sources: 15,088 Total siz	e: 5.17 GiB	1 70 140	Y	1 70 100	A 6.4	0.6.4.4	A 6.411	Da	~





Ray tracing resources are colored in all views as

- Timeline view
- Heap overview
- Resource overview
- Allocation overview
- Allocation explorer
- Resource List
- Resource Details



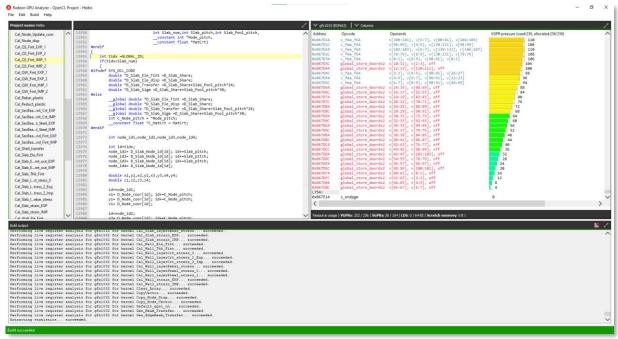




AMDJ RADEON GPU Analyzer



- Compile and analyze shaders and kernels for the RDNA3[™] architecture (gfx1100, RX 7000 series GPUs)
 - Without a physical RDNA3[™] card installed on your system
- New VGPR pressure GUI
 - Displays highest VGPR pressure and helps you identify where to focus your optimization
- DirectX®11 mode (-s dx11) is now an offline mode
 - Works regardless of the installed driver. Support in this mode was limited to VS, PS and CS
- Updates to Vulkan[®] and OpenGL[®] backend components
- OpenCL[™] mode now uses an updated version of AMD's LLVMbased Lightning Compiler







- Displays number of VGPRs used and allocated
- Utilization bars that visualize the VGPR usage and the VGPR block utilization

💙 gfx1032 (RDNA2) 📗 🔪	Columns	/
Opcode	Operands	VGPR pressure (used:252, allocated:256/256)
global_store_dword>	2 v[6:7], v[90:91], off	104
global_store_dword>	2 v[8:9], v[114:115], off	100
v_fma_f64	v[0:1], v[0:1], v[86:87], v[22:23	96
v_fma_f64	v[2:3], v[2:3], v[86:87], v[18:19	94
v_fma_f64	v[4:5], v[4:5], v[86:87], v[44:45	92
<pre>global_store_dword></pre>	2 v[10:11], v[64:65], off	88
<pre>global_store_dword></pre>	2 v[12:13], v[48:49], off	84
<pre>global_store_dword></pre>	2 v[14:15], v[78:79], off	80
<pre>global_store_dword></pre>	2 v[16:17], v[66:67], off	76
<pre>global_store_dword></pre>	2 v[20:21], v[92:93], off	72
<pre>global_store_dword></pre>	2 v[24:25], v[46:47], off	68
<pre>global_store_dword></pre>	2 v[26:27], v[68:69], off	64
<pre>global_store_dword></pre>	2 v[28:29], v[88:89], off	60
<pre>global_store_dword></pre>	2 v[30:31], v[98:99], off	56
<u> </u>	2 v[32:33], v[70:71], off	52
<u> </u>	2 v[34:35], v[80:81], off	48
<u> </u>	2 v[36:37], v[82:83], off	44
<u> </u>	2 v[38:39], v[72:73], off	40
<u> </u>	2 v[40:41], v[96:97], off	36
<u> </u>	2 v[42:43], v[102:103], off	32
<u> </u>	2 v[50:51], v[74:75], off	28
<u> </u>	2 v[52:53], v[94:95], off	24
<u> </u>	2 v[54:55], v[100:101], off	20
	2 v[56:57], v[76:77], off	16
<u> </u>	2 v[58:59], v[0:1], off	12
·	2 v[60:61], v[2:3], off	8
global_store_dword>	2_v[62:63], v[4:5], off	N





- Displays number of VGPRs used and allocated
- Utilization bars that visualize the VGPR usage and the VGPR block utilization
- Tooltips of live VGPRs per instruction with reduction hints

	/	
	VGPR pressure (used:252, allocated:256/256)	•
7] 3] 9]	104 100 98 96 94 88 88 84 80 76	
	76 used, 80 allocated out of 256 VGPRs. Reduce VGPR consumption by 4 to save 8 VGPRs (HW allocates VGPRs in blocks of 8) 60 56 52 48 44 40 36 32 28 24	
	20	





- Displays number of VGPRs used and allocated
- Utilization bars that visualize the VGPR usage and the VGPR block utilization
- Tooltips of live VGPRs per instruction with reduction hints
- F4 shortcut will cycle through the lines with maximum VGPRs. Shift+F4 will do the same in reverse order

💙 gfx1032 (RD	NA2) 💙 Columns	1
Opcode	Operands	VGPR pressure (used:236, allocated:256/256)
v_fma_f64	v[144:145], v[26:27], v[198:199], v[216:217]	236
v_fma_f64	v[216:217], v[52:53], v[198:199], v[218:219]	236
v_fma_f64	v[172:173], v[4:5], v[198:199], v[172:173]	234
s_waitcnt	vmcnt(10)	232
v_fma_f64	v[198:199], v[26:27], v[202:203], v[220:221]	234
v_fma_f64	v[220:221], v[72:73], v[130:131], v[228:229]	234
s_waitcnt	vmcnt(8)	232
v_fma_f64	v[178:179], v[4:5], v[210:211], v[178:179]	232
v_fma_f64	v[228:229], v[52:53], v[210:211], v[246:247]	234
v_fma_f64	v[218:219], v[52:53], v[202:203], v[222:223]	234
v_mul_f64	v[126:127], v[200:201], v[174:175]	234
v_fma_f64	v[182:183], v[4:5], v[202:203], v[182:183]	234
v_fma_f64	v[202:203], v[68:69], v[130:131], v[224:225]	234
v_fma_f64	v[222:223], v[70:71], v[134:135], v[236:237]	234
v_fma_f64	v[224:225], v[74:75], v[134:135], v[238:239]	234
v_fma_f64	v[204:205], v[194:195], 0, v[204:205]	230
v_mul_f64	v[134:135], v[124:125], s[20:21]	232
v_fma_f64	v[168:169], v[4:5], v[208:209], v[168:169]	232
v_fma_f64	v[142:143], v[8:9], v[194:195], v[142:143]	232
v_fma_f64	v[192:193], v[50:51], v[184:185], v[192:193]	230
v_mul_f64	v[124:125], v[148:149], v[124:125]	230
v_fma_f64	v[120:121], -v[120:121], v[188:189], v[232:233]	228
v_mul_f64	v[138:139], v[48:49], v[190:191]	228
v_mul_f64	v[2:3], v[96:97], v[2:3]	228
v_mul_f64	v[190:191], v[6:7], v[190:191]	228
v_fma_f64	v[94:95], v[8:9], v[184:185], v[94:95]	228
v_fma_f64	v[130:131], v[80:81], v[130:131], v[234:235]	228
v_mul_f64	v[0:1], v[132:133], v[0:1]	226
s_waitcnt	vmcnt(7)	226
v fma f64	v[148:149], v[52:53], v[196:197], v[250:251]	228





RADV & STEAM DECK SUPPORT

- RADV is a Vulkan[®] driver for AMD Radeon[™] GPUs written for Linux[®] by the community as part of the Mesa project
- Radeon tools support in RADV enabled via environment variables for Steam Deck device
 - RGP support
 - **RADV_THREAD_TRACE**= frameindex
 - **RADV_THREAD_TRACE_TRIGGER**=*filename*
 - "RGP Capture" button in the SteamOS Devkit client
 - RRA support
 - · Added in Mesa 22.3
 - **RADV_RRA_TRACE**= frameindex
 - **RADV_RRA_TRACE_TRIGGER**=*filename*
 - RMV support
 - Added in Mesa 23.0 (also requires Linux kernel 5.10 or newer)
 - **MESA_VK_MEMORY_TRACE**= frameindex
 - MESA_VK_MEMORY_TRACE_TRIGGER=filename





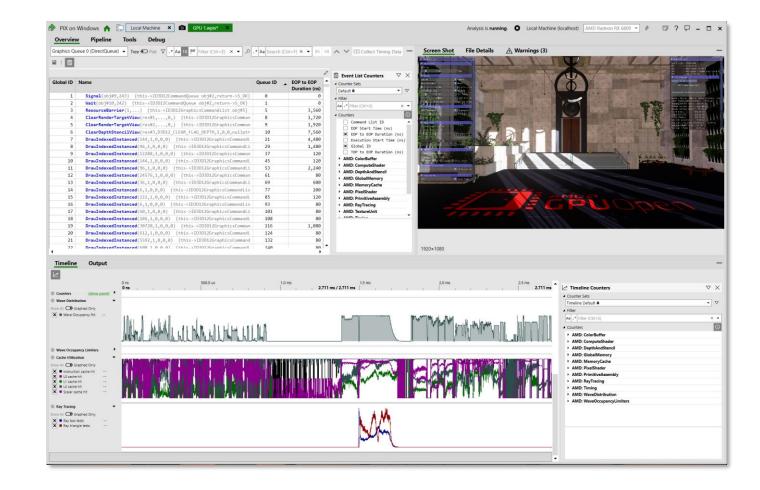
THIS PRESENTATION

- Part 1: Overview of the Radeon[™] Developer Tool Suite
 - What is in it
 - What's new since GDC 2022
 - Dive into new features and improvements
 - RADV & Steam Deck support
- Part 2: Collaboration with external tool developers
 - Microsoft[®] PIX on Windows
 - RenderDoc
 - GFXReconstruct



MICROSOFT PIX ON WINDOWS

- Debugging and performance analysis
- Event List Counters
- High frequency counter graphs
 - Wave distribution
 - Wave occupancy limiters
 - Cache utilization NEW!
 - Scalar cache hit
 - Instruction cache hit
 - LO/L1/L2 cache hit
 - Raytracing NEW!
 - Ray box tests
 - Ray triangle tests
- <u>https://devblogs.microsoft.com/pix/</u>
- DirectX Discord server: discord.gg/directx





RENDERDOC - OVERVIEW

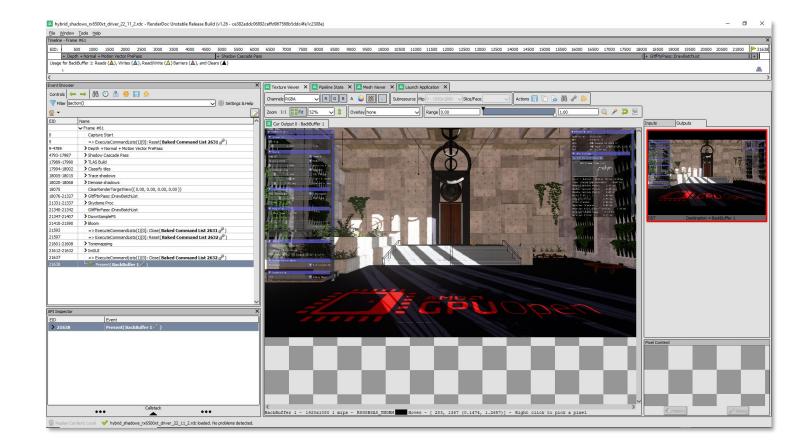
- RenderDoc is a free MIT licensed stand-alone graphics debugger
- Allows quick and easy single-frame capture and detailed introspection
- Applications using Vulkan, D3D11, OpenGL & OpenGL ES or D3D12 across Windows, Linux, Android
- RDNA3 GPA Counter support NEW!

																				- 0
	Tools Help																			
meline - Frame	e #28																			
ID:	1000	2754	4000	5000	6000	7000	8000	9000	10000	11000	12000	13000	14000	15000	16000	17000	18000	19000	20000	2: 2
		Notion Vector PrePass			v Cascade Pass												+ Gltfl	PbrPass::DrawBa	atchList	
sage for Back	kBuffer 0: Rea	ds (🛕), Writes (🛕), Read/W	rite (🛆) Barriers (,	A), and Clears	(▲)															
¥.																				
ent Browser									>	Textu 🗙	Pipeli	× 🔼 Launc	× Reso	X Perfor	mance X	Statis X	(11) Errors.	🗙 🗖 Mesh	× 🗖 Hyt	oridShadows I
ontrols 🦛	→ 船	🕥 🐧 🐫 🖪 🏚										1			1		-			
Filter Sacti		- 1010 - 0.0						V 🔅 Set	tinor & Halo	Channels RG	BA 🗸	RGB	A 🥥 💹	Y Subreso	urce Mip 0 - 19	20x1080 VS	lice/Face	 Actio 	ns 🔚 🗈 🛛	s) 88 °
	10110								ungs och leip		69									
•									1	Zoom 1:1	🖥 🖥 Fit 36%	_ ℓ	Overlay None							
0		Name							^	Range 0.00		T	·	1.00		1 🔍 🎤 🧋				
		Frame #28								Kange [0.00				∆[1.00						
	0	Capture Start								Cur Outpi	ut 0 - BackBuffer	r 0							Inputs	Output
	1	=> ExecuteCommandList		ked Command	List 2536 🖉)					WOMENLIND	1000	1 1	and the	T	1	1	* ANILA T L	1		
789	1-618	> Depth + Normal + Motion	Vector PrePass							No. of Concession, No.		of the second second	· ·	X		and the state	C			
	618-2461	Shadow Cascade Pass								Personal Property in	-		and the	A	1	We many and				
89-17990	3464	✓ TLAS Build		"						a surface and			- · · · ·	(0)			ality of a contra			
90	2461	BuildRaytracingAcceler Classify tiles	ationStructure({ A	SButter::m_pl	Butter 💣 , No R	esource 🖉 ,	ASButter::m	_pButter () })		Constant Section	a der		a second		teres and		and the second	(Then		
94-18002	2463-2464	Trace shadows								A Sugar	Cargo a stars			1-1	14 a. a. a.		THE R			
	2465-2471	Denoise shadows								C Productional and	107/11	Laugerran The State						J ARC S	SRC	Source
075	2405-24/1	ClearRenderTargetView(0.00.0.00.0.00	0.00.33						LIME	Thursday.	MAT					Constant and the	133		
		> GitfPbrPass::DrawBatchL		0.00 /)						1. Provide to	ALC: NO	-			Care and	Sell		Care Care		
331-21337	2172 2052	Skydome Proc	ac								(bitter %				· · · ·		* mudu	mm		
340-21342		GltfPbrPass::DrawBatchL	ist							101	Annal Care	E. Maria			1 1 1 1 1			VIENT		
	2834-2839	> DownSamplePS								1	States and a				E IN	1 1		The second	Pixel Conte	vt
		> Bloom								Farmers	KOM	the start	A			K bai	and hundred	THE	FIXELCOILLE	
593	2854	=> ExecuteCommandList	s(1)[0]: Close(Bak	ed Command	List 2536 (2)					A definition of the										
597	2854	=> ExecuteCommandList								R. Sectors of a	Total T									
		•							>	1.5%										_
I Inspector													VED	A PHILE						
		12							,				VE (0						
0	_	Event				_		_	_											
21638		Present(BackBuffer 0	-)																	
																				1000
																	_			
																_				
										<				_		_		>		
										Destablishes	0 1000	1000 1	- R8G8B8A8	INCOM	Horrow	75 625 10		071		
		•••		Callstack							k to pick a		- KOGODOHO	ONORH	HOVEL -	13, 023 11	0.0291, 0.2/	0/1 =	(Histo	KY 25



RENDERDOC – WORK IN PROGRESS

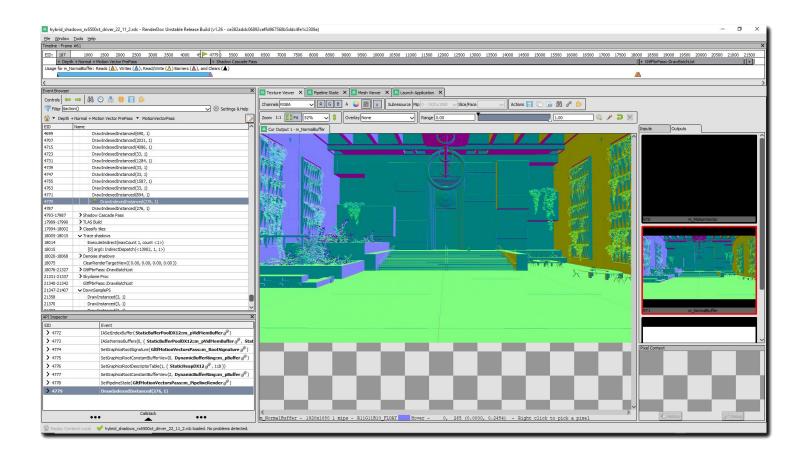
- Collaboration between AMD and Baldur Karlsson to support capture of DXR applications
- DXR Support
 - Capture/replay DXR workload





RENDERDOC – RGP INTEROP

- Generate RGP profile from capture file
- Profile opened in RGP
- Select event in RenderDoc
- Jump to same event in RGP
- Select event in RGP
- Jump to same event in RenderDoc





GFXRECONSTRUCT + D3D12 + DXR

- What is it?
 - Graphics workload capture/replay software suite
 - Replacement for vktrace
 - Command line based
 - Added D3D12 + DXR support in joint venture between LunarG and AMD
- Highlights
 - Supports ray tracing
 - Fully open-source
 - Extensible software architecture
 - Can capture/replay thousands of frames
 - Lightweight





GFXRECONSTRUCT + D3D12 + DXR

- Provided components
 - <u>Writing</u>: DLLs to capture a GPU workload
 - <u>Reading</u>: executables to replay and process the captured workload
- Main tools: gfxrecon-replay.exe and gfxrecon-optimize.exe
- Other tools, that can:
 - Display the capture file metadata
 - Change the compression algorithm
 - Generate a human readable API dump
- How to get it
 - Source code and binaries: <u>https://github.com/LunarG/gfxreconstruct</u>
 - Blog: https://gpuopen.com/learn/amd-lunarg-gfxreconstruct-dx12-dxr



- Use cases
 - Open-source toolkit
 - Profiling specific frames
 - Debugging visual artifacts
 - Stress testing
 - Experimentation platform





Chris Hesik

christopher.hesik@amd.com

Can Alper can.alper@amd.com

Radeon Developer Tool Suite

https://gpuopen.com/tools/

Tool Suite Videos

https://gpuopen.com/videos/



DISCLAIMER

GENERAL DISCLAIMER

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's products are as set forth in a signed agreement between the parties or in AMD's Standard Terms and Conditions of Sale. GD-18

© 2023 Advanced Micro Devices, Inc. All rights reserved. AMD, the AMD Arrow logo, Radeon and combinations thereof are trademarks of Advanced Micro Devices, Inc. DirectX is a registered trademark of Microsoft Corporation in the US and other jurisdictions. Linux is a registered trademark of Linux Torvalds. OpenGL[®] and the oval logo are trademarks or registered trademarks of Hewlett Packard Enterprise in the United States and/or other countries worldwide. OpenCL is a trademark of Apple, Inc. used by permission from The Khronos Group. SPIR, SPIR-V and the SPIR logo are trademarks of the Khronos Group Inc. Vulkan and the Vulkan logo are registered trademarks of the Khronos Group Inc. Windows is a registered trademark of Microsoft Corporation in the US and other jurisdictions. Other product names used in this publication are for identification purposes only and may be trademarks of their respective companies.



GPUOpen

together we advance_

AMD



ANY QUESTIONS?

Part 1: Overview of the Radeon[™] Developer Tool Suite

- What is in it
- What's new since GDC 2022
- Dive into new features and improvements
- RADV & Steam Deck support

Part 2: Collaboration with external tool developers

- Microsoft[®] PIX on Windows
- RenderDoc
- GFXReconstruct

Chris Hesik

christopher.hesik@amd.com

Can Alper can.alper@amd.com

Radeon Developer Tool Suite

https://gpuopen.com/tools/

Tool Suite Videos

https://gpuopen.com/videos/

