

Trace-based profiling on AMD GPUs

Bas Nieuwenhuizen <bas@basnieuwenhuizen.nl>
2020-09-16



Profiling : perfcounters + barriers

Measurement:

- Timestamps
- GPU perfcounters

Barriers => No inter-draw effects

Barrier

Sample top

Draw

Sample bottom

Barrier

Profiling : perfcounters without barriers

Sample top
Draw
Sample bottom

Sample top
Draw
Sample bottom

- vkCmdDrawIndexed(102, 10)	40.32
- vkCmdDrawIndexed(102, 10)	43.96
- vkCmdDrawIndexed(60, 10)	197.80
- vkCmdDrawIndexed(60, 2)	211.84
- vkCmdDrawIndexed(102, 46)	807.92
- vkCmdDrawIndexed(102, 117)	952.28
- vkCmdDrawIndexed(102, 19)	1153.60
- vkCmdDrawIndexed(102, 6)	1130.48
- vkCmdDrawIndexed(120, 18)	1127.28
- vkCmdDrawIndexed(120, 106)	959.96
- vkCmdDrawIndexed(120, 2)	959.48
- vkCmdDrawIndexed(9, 29)	361.24
- vkCmdDrawIndexed(9, 9)	216.44
- vkCmdDrawIndexed(9, 17)	11.28
- vkCmdDrawIndexed(9, 13)	12.24
- vkCmdDrawIndexed(9, 34)	11.80

Thread traces

Log:

- Subgroup start/stop
- Draw start/stop
- Driver annotations
- Register writes
- Events

```
$ RADV_THREAD_TRACE_TRIGGER=/tmp/sqtt %command%
```

```
$ touch /tmp/sqtt # Traces the next frame
```

```
# Frame in /tmp/*.rgp
```



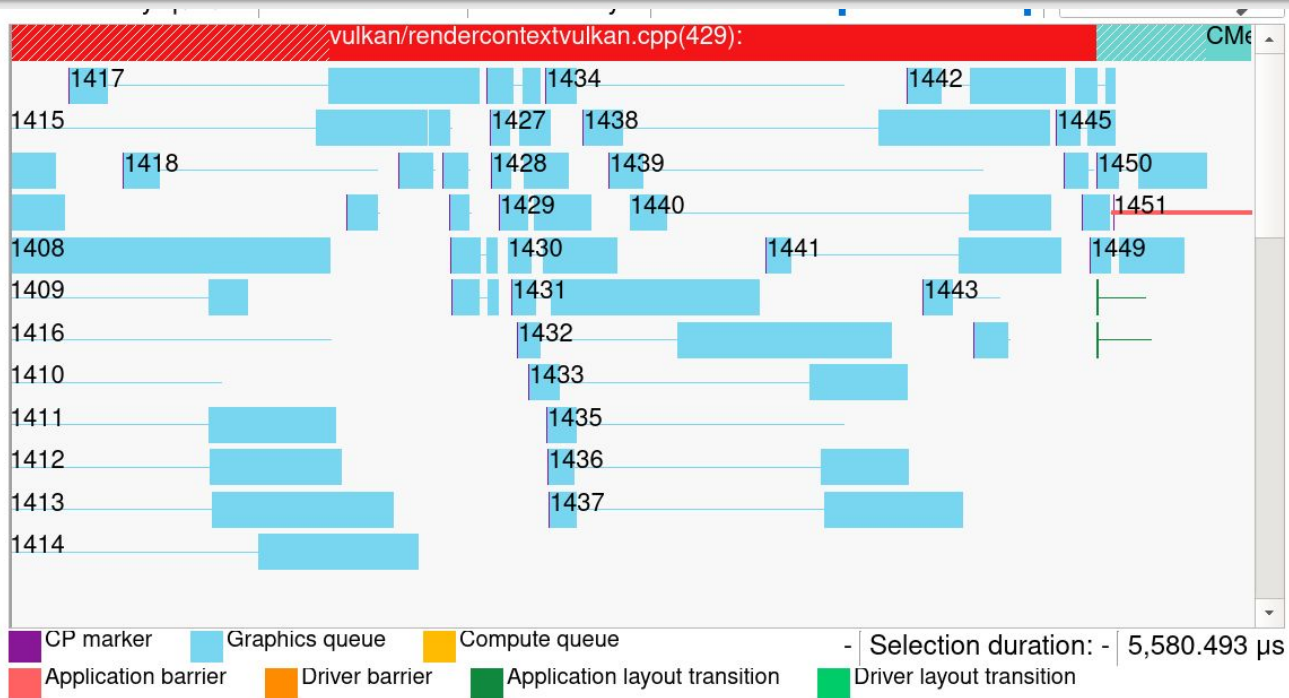
Radeon GPU Profiler

<https://gpuopen.com/rgp/>

RGP Wavefront/Subgroup view



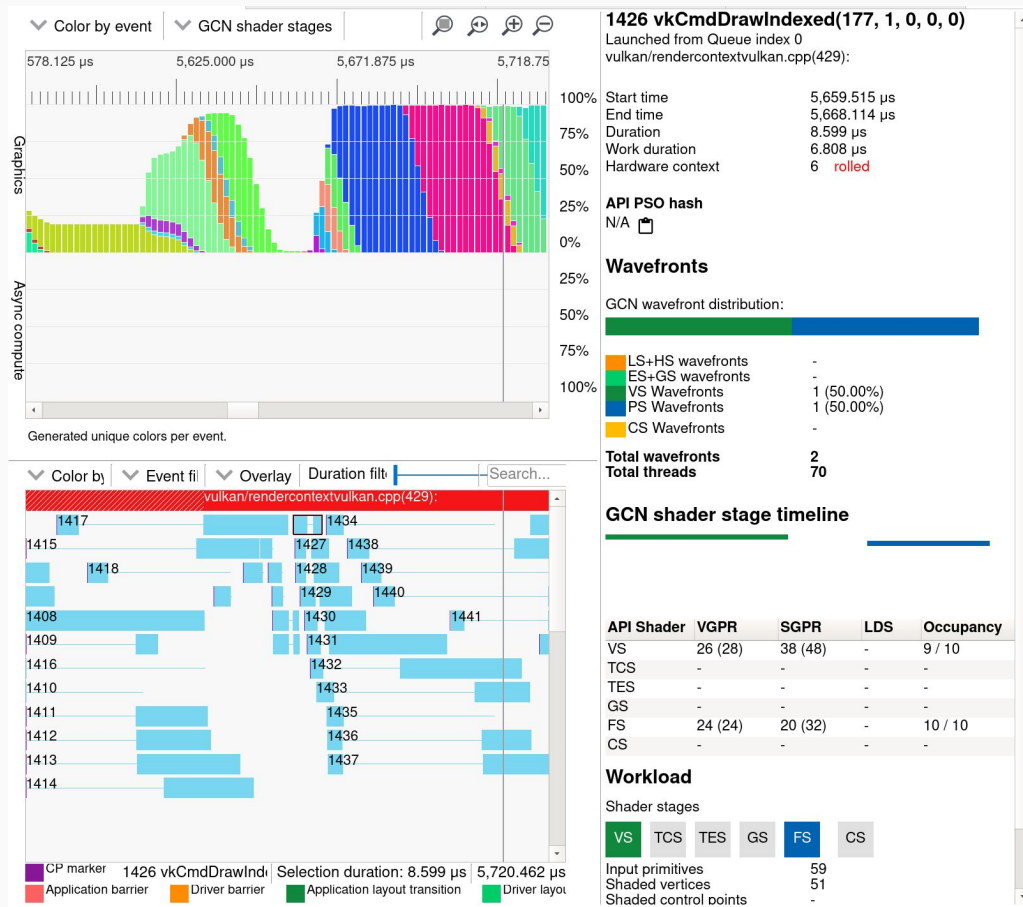
Draws



Occupancy

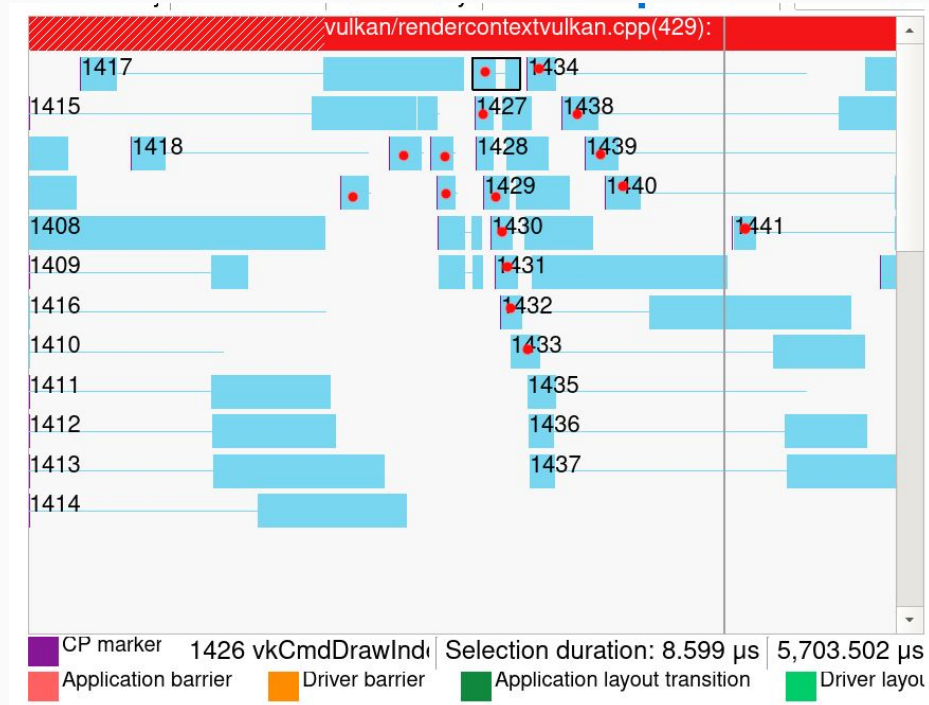


Full picture



Optimization Opportunity: State changes

- Max 7 draw states in flight
- Frequent state changes cause stalls



Optimization Opportunity: Barriers

Vega 64 full occupancy = 164k invocations

-> GPU stalls are expensive

Event Numbers	Duration	Drain Time	Stalls	De Hi: DC FN Fa Init Mask De Re De De Eli RAM	Invalidated	Flushed	Type	Reason
0	0.016 μ s	0.016 μ s					APP	
2	35.851 μ s	8.714 μ s	FULL				APP	
3	4.391 μ s	4.391 μ s	FULL				DRIVER	Unknown driver barrier.
4.6	17.291 μ s	17.291 μ s	FULL CS		✓ L1 DB	DB	APP	
6	3.603 μ s	17.291 μ s			✓		APP	
8	1.772 μ s	0.000 μ s			K L1 DB	DB	DRIVER	Before CS depth/stencil clear
10	4.157 μ s	0.000 μ s	CS		K L1		DRIVER	After CS depth/stencil clear.
14	0.573 μ s	0.000 μ s			K L1 L2 CB DB	L2 CB DB	APP	

Optimization Opportunity: Barriers



Overhead

- Typical trace for AAA game is 0.5-1 GiB
 - ~10% bandwidth overhead on top discrete GPUs
-
- Slower than timestamps without barriers
 - Faster than replay tools

Bonus: Instruction Level Profiling

		Hit count	struction cost (%)	Average latency
4	v_lshl_add_u32 v0, s3, 8, v0	264	0.22	84 clk
5	s_waitcnt lgkmcnt(0)	264	0.17	68 clk
6	s_buffer_load_dword s2, s[12:15], 0xc	264	0.01	4 clk
7	s_waitcnt lgkmcnt(0)	264	0.72	283 clk
8	v_cmp_gt_u32_e32 vcc, s2, v0	264	0.01	4 clk
9	s_and_saveexec_b64 s[2:3], vcc	264	0.06	23 clk
10	s_cbranch_execz BB0_7	264	0.01	4 clk
11	BBF0_0 s_load_dwordx4 s[4:7], s[0:1], 0x0	264	0.02	8 clk
12	v_lshlrev_b32_e32 v4, 5, v0	264	0.07	29 clk
13	s_mov_b32 s0, 0x35eae18b	264	0.02	7 clk
14	s_waitcnt lgkmcnt(0)	264	12.98	5,07
15	buffer_load_dwordx4 v[5:8], v4, s[4:7],	264	0.01	4 clk
16	s_buffer_load_dword s10, s[12:15], 0x8	264	12.70	4,962
17	s_buffer_load_dwordx2 s[8:9], s[12:15],	264	0.01	4 clk
18	s_waitcnt vmcnt(0) lgkmcnt(0)	264	6.78	2,649 clk
19	v_sub_f32_e32 v2, s10, v6	264	0.01	4 clk
20	v_sub_f32_e32 v3, s9, v5	264	0.01	4 clk
21	v_mul_f32_e32 v1, v2, v2	264	0.01	4 clk
22	v_mac_f32_e32 v1, v3, v3	264	0.01	4 clk
23	v_sqrt_f32_e32 v9, v1	264	0.04	16 clk
24	v_mul_f32_e32 v1, v1, v9	264	0.01	4 clk
25	v_rcp_f32_e32 v9, v1	264	0.04	16 clk
26	v_or_b32_e32 v1, 8, v4	264	0.01	4 clk
27	v_mul_f32_e32 v3, v3, v9	264	0.01	4 clk
28	v_mul_f32_e32 v9, v2, v9	264	0.01	4 clk
29	v_mad_f32 v2, -v3, s0, v7	264	0.01	4 clk
30	v_mad_f32 v3, -v9, s0, v8	264	0.01	4 clk
31	v_mad_f32 v5, s8, v2, v5	264	0.01	4 clk

Log all instructions from 1 Compute Unit

TODO

Streaming performance counters

- Occupancy isn't utilization
- Can sample perfcounters each ~ 100 ns
- No RGP support

Q & A