AMDGPU kernel driver change summary:

Impact: Radeon RX 7000 series, Radeon RX 6000 series, Radeon RX 5000 series

Summary:

- Set UNORD_DISPATCH bit in the cp_hqd_pq_control field in the memory queue descriptor (MQD) for compute queues on GFX10 and 11 based CPUs.
- The AMD GPU has a unit called Command Processor (CP) in charge of driving work based on information provided by the driver. The MQD is an object in memory that describes the configuration of a compute queue. The firmware reads the state out of the MQD and puts it into a hardware slot when the queue is scheduled.
- When a queue is de-scheduled, the firmware saves the queue state from the hardware slot back to the MQD. For compute workloads, there is a handshake between the CP and the underlining hardware when dispatching new kernels/waves to ensure that the required resources are available. Those dispatches can be committed in-order or out-of-order.
- In older hardware, the out-of-order setting conflicted with fast context switching and was disabled. That setting was inherited by newer hardware, even though more recent designs do allow the out-of-order setting to work properly. During dispatch it is also possible to pre-allocate resources associated with certain waves. The combination of an unbounded pre-allocation with in-order dispatches enabled could result in deadlock due to a lack of resources over time depending on use case.
- Changing to out-of-order dispatches on newer hardware can prevent this deadlock from occurring while still allowing fast context switching. The driver sets the global enable bit for this feature in the MQD to allow individual dispatches to set in order or out of order. These changes need to be coupled with updated CP firmware which properly enables out of order dispatches when processing AQL dispatch packets.

CP firmware change summary:

Impact: Radeon RX 7000 series, Radeon RX 6000 series, Radeon RX 5000 series

Summary:

- The AMD GPU has a unit called Command Processor (CP) in charge of driving work based on information provided by the driver. For compute workloads, there is a handshake between the CP and the underlining hardware when dispatching new kernels/waves to ensure that the required resources are available.
- Those dispatches can be committed in-order or out-of-order. In older hardware, the out-of-order setting conflicted with fast context switching and was disabled. That setting was inherited by newer hardware (Navi3x onwards), even though more recent designs do allow the out-of-order setting to work properly. During dispatch it is also possible to pre-allocate resources associated with certain waves. The combination of an unbounded pre-allocation with in-order dispatches enabled could result in deadlock due to a lack of resources over time depending on use case.
- Changing to out-of-order dispatches on newer hardware can prevent deadlocks from occurring while still allowing fast context switching.

MES FW change summary:

Impact: Radeon RX 7000 series, Radeon RX 6000 series, Radeon RX 5000 series

Summary:

- The Micro-Engine Scheduler (MES) is a hardware engine used by AMD Graphics IP, GFX, for GPU workload scheduling. The MES FW interacts with the kernel driver via a ring buffer to schedule user queues to the hardware queues of each engine.

- MES process context gets corrupted when the driver reuses the process context memory after calling MES API MESAPI__MISC.MESAPI_MISC__SET_SHADER_DEBUGGER to enable shader debugger but not adding any queues. Since queues are not added to the process, the debugger process will not be removed from the process linked list, and when the driver reuses the memory and overwrites the fields, the linked list will become corrupted.

- The solution is add a new flag to the MES API for driver to set to remove the shader debugger process from the mes process list. This flag should only be used if queues were not added to the process.

Observations under investigation:

- Unit test cases using HIP may fail at higher optimization levels in both single and multi-GPU mode
- Potential memory leak observed when running ROCm-Bandwidth and GPUBurn-HIP test simultaneously with 2 GPUs for prolonged time
- Intermittent error message of tm_bo_delayed_delete [amdttm] hogged CPU message may be observed in dmesg