Device-Generated Commands in Vulkan (VK_EXT_device_generated_commands)

Ricardo Garcia

XDC 2024 – October 10 - Montreal

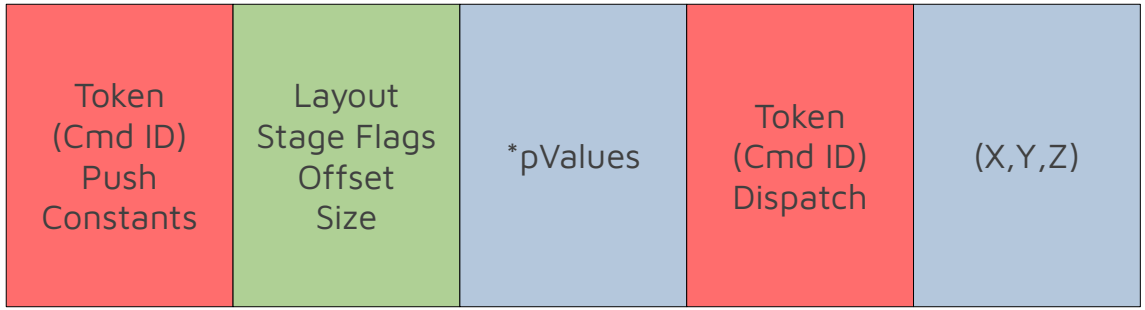
igalia

What are Device-Generated Commands?

- One step ahead of indirect draws and dispatches
- One step behind work graphs
- Allows implementations to read command sequences from a regular buffer instead of a command buffer
- That buffer could be filled from the GPU to achieve GPU-driven rendering

Naïve CPU-based Approach

1) vkCmdPushConstants(layout, stageFlags, offset, size, pValues)
2) vkCmdDispatch(x, y, z)

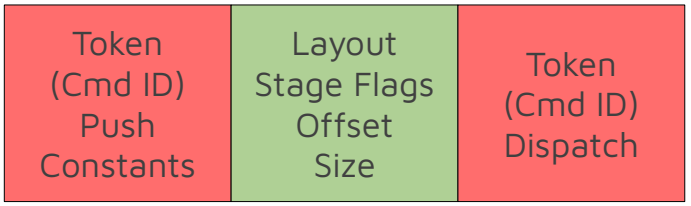


VK_EXT_device_generated_commands

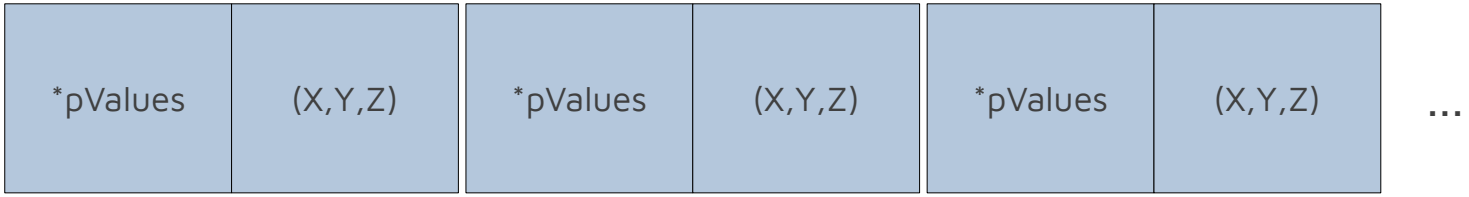
- VkIndirectCommandsLayoutEXT

  1) vkCmdPushConstants

  2) vkCmdDispatch



- Buffer contains a number of fixed-size sequences and each follows the layout



Restriced Command Selection

VK_INDIRECT_COMMANDS_TOKEN_TYPE_EXECUTION_SET_EXT
VK_INDIRECT_COMMANDS_TOKEN_TYPE_PUSH_CONSTANT_EXT
VK_INDIRECT_COMMANDS_TOKEN_TYPE_SEQUENCE_INDEX_EXT

VK_INDIRECT_COMMANDS_TOKEN_TYPE_INDEX_BUFFER_EXT
VK_INDIRECT_COMMANDS_TOKEN_TYPE_VERTEX_BUFFER_EXT
VK_INDIRECT_COMMANDS_TOKEN_TYPE_DRAW_INDEXED_EXT
VK_INDIRECT_COMMANDS_TOKEN_TYPE_DRAW_EXT
VK_INDIRECT_COMMANDS_TOKEN_TYPE_DRAW_INDEXED_COUNT_EXT
VK_INDIRECT_COMMANDS_TOKEN_TYPE_DRAW_COUNT_EXT

VK_INDIRECT_COMMANDS_TOKEN_TYPE_DISPATCH_EXT

VK_INDIRECT_COMMANDS_TOKEN_TYPE_TRACE_RAYS2_EXT

VK_INDIRECT_COMMANDS_TOKEN_TYPE_DRAW_MESH_TASKS_NV_EXT
VK_INDIRECT_COMMANDS_TOKEN_TYPE_DRAW_MESH_TASKS_COUNT_NV_EXT
VK_INDIRECT_COMMANDS_TOKEN_TYPE_DRAW_MESH_TASKS_EXT
VK_INDIRECT_COMMANDS_TOKEN_TYPE_DRAW_MESH_TASKS_COUNT_EXT

Indirect Execution Sets

- A group of similar pipelines or shader objects
- All state must be identical (only shaders change)
- Each pipeline/shader has an index in the set
- The IES to use is specified beforehand and the DGC buffer contains indices into the set

Quick How-To

1) Create the commands layout, and IES if needed (VkIndirectCommandsLayoutEXT, VkIndirectExecutionSetEXT)
2) Establish the maximum number of sequences
3) Query the required preprocess buffer size (vkGetGeneratedCommandsMemoryRequirementsEXT)
4) Allocate DGC buffer and preprocess buffer
5) Record commands and state almost normally (including work that fills the DGC buffer)
6) Dispatch work with vkCmdExecuteGeneratedCommandsEXT
7) If using explicit preprocessing (e.g. Proton does it to improve performance):
   a) Use a separate command buffer for it
   b) Pass the main command buffer in as state
   c) Call vkCmdPreprocessGeneratedCommandsEXT and submit this work first, synchronizing with vkCmdExecuteGeneratedCommandsEXT

Thanks for watching!

Join us!

https://www.igalia.com/jobs