



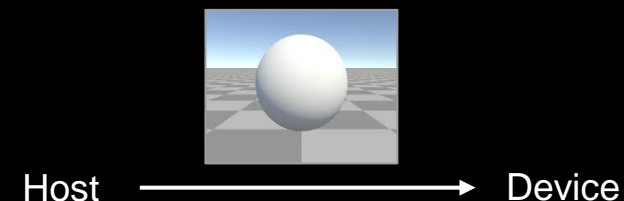
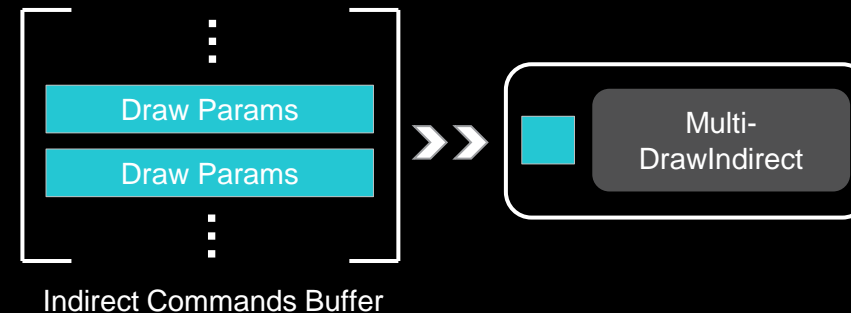
GPU-driven Rendering in Vulkan

Ken Shanyi Zhang

GPU-driven Rendering

Motivation

- Reduce round trips to the host
 - More flexible indirect commands
- Launch work on the GPU
 - Specify dependency of GPU works
 - Automate state changes
 - GPU culling
- Improve GPU programmability
 - Global expressiveness



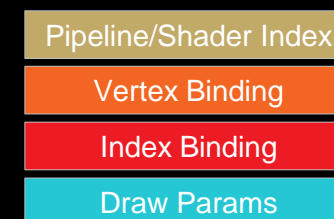
VK_EXT_device_generated_commands

A new Vulkan extension for such pursuit

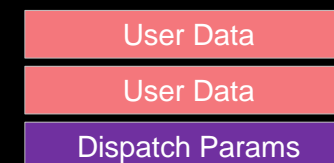
VK_EXT_device_generated_commands

In a nutshell

- Extend indirect commands
 - It offers the flexibility to customize indirect commands layout
- Preprocessing
 - Overlap commands generation with execution
 - Memory consumption control
- Expressiveness
 - Shader inputs can also be handled in indirect fashion
 - Indirect pipeline/shader switching



or

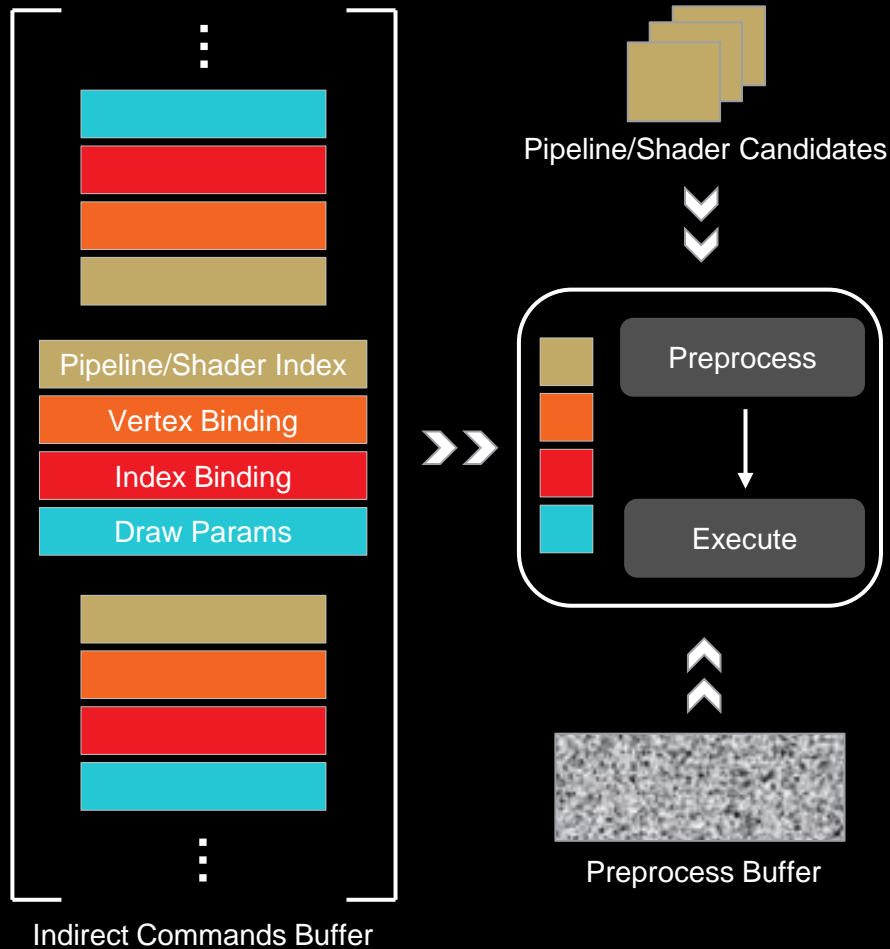


⋮

Indirect Commands Layout

VK_EXT_device_generated_commands

In a nutshell

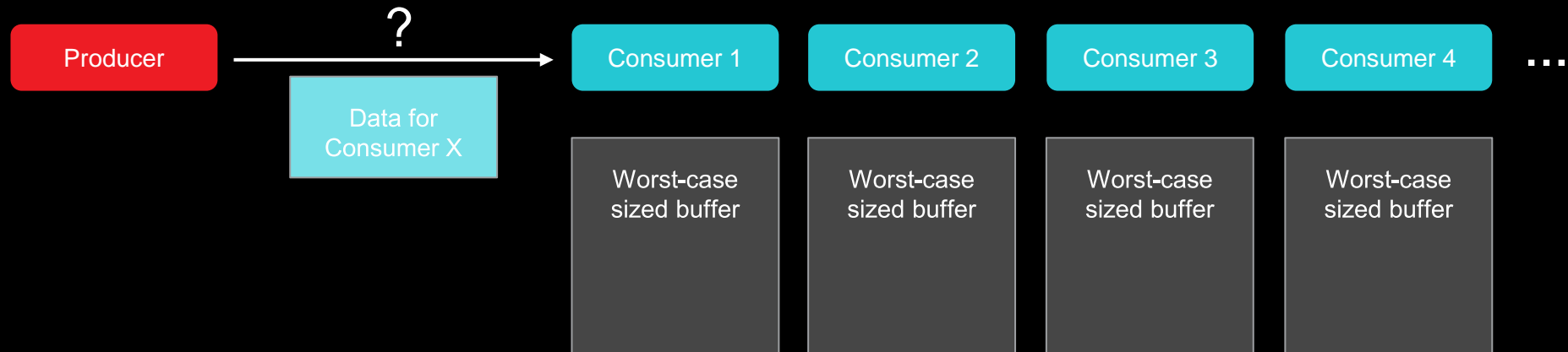


- Procedures

- Create a pipeline/shader “collection”
- Determine a command layout
- Memory allocation
- Preprocessing (optional)
- Execution

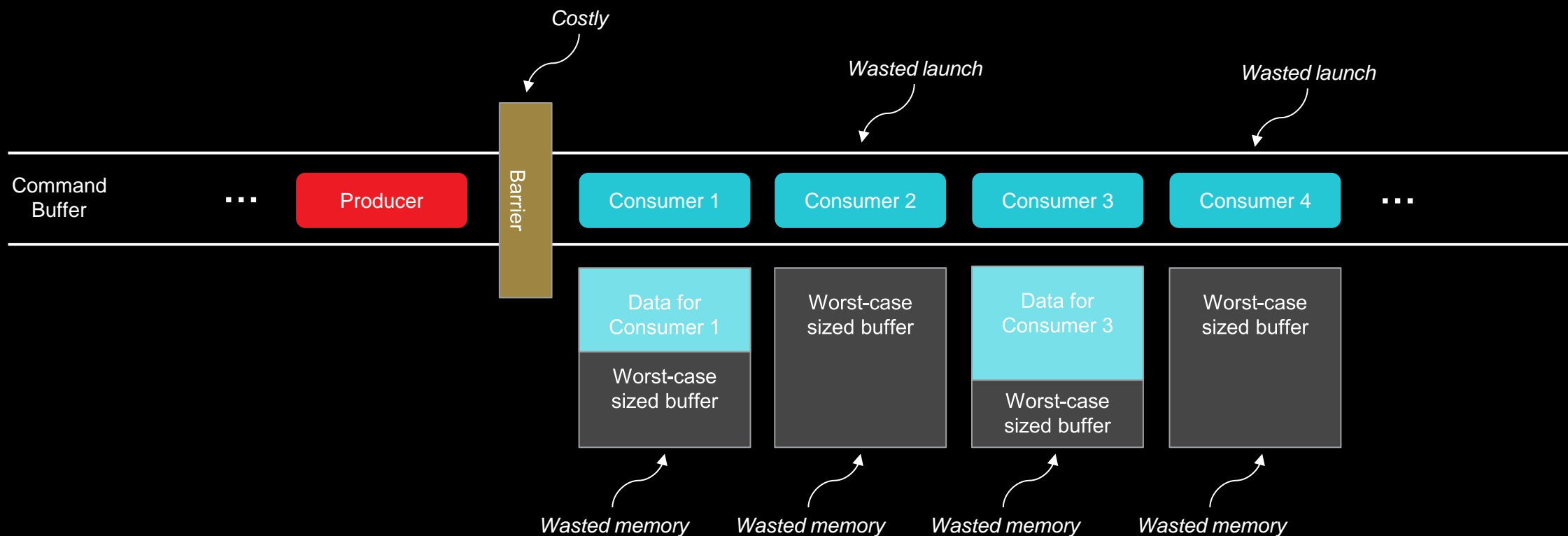
VK_EXT_device_generated_commands

What's the problem?



VK_EXT_device_generated_commands

What's the problem?



VK_EXT_device_generated_commands

What's the problem?

- Wasted launch
- Wasted memory
- Indirect pipeline/shader switching
- Very implementation-dependent

What if ...

Expressiveness

The GPU could support more advanced algorithms and complex dependency relations.

Memory Management

The GPU could manage memory consumption.

Optimization

The GPU runtime could optimize for better performance.

Self-scheduling

The GPU could launch appropriate work at appropriate time.

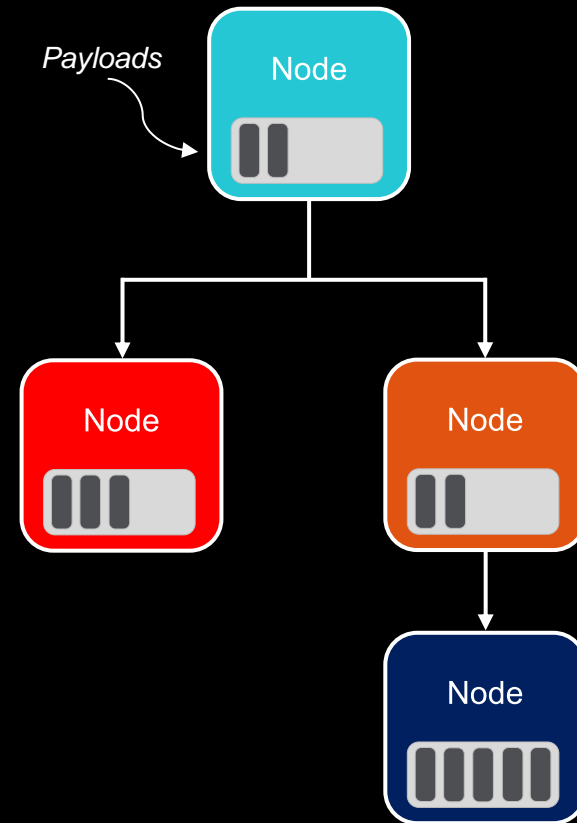
VK_AMD_X_shader_enqueue

The extension that represents the future

VK_AMD_X_shader_enqueue

In a nutshell

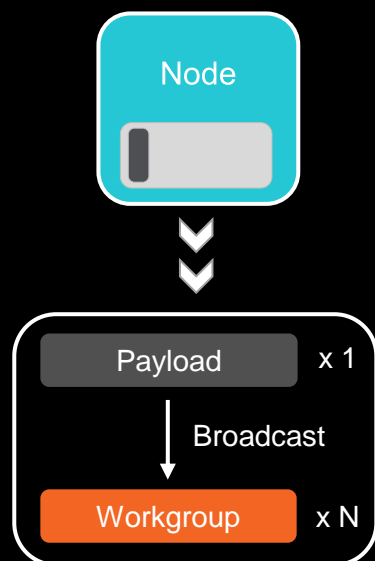
- A brand-new data flow model
 - Shaders are represented as nodes
 - Many nodes with a virtual queue for each
 - Payloads flow between nodes and get enqueued
- Nodes launch when it is appropriate
 - There is enough pending payloads
 - It also depends on GPU, driver and the graph itself
 - Node merge, reordering, sorting in runtime



VK_AMD_X_shader_enqueue

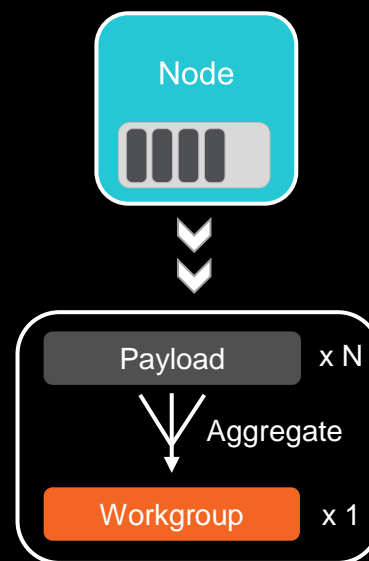
In a nutshell

- You can select launch mode:



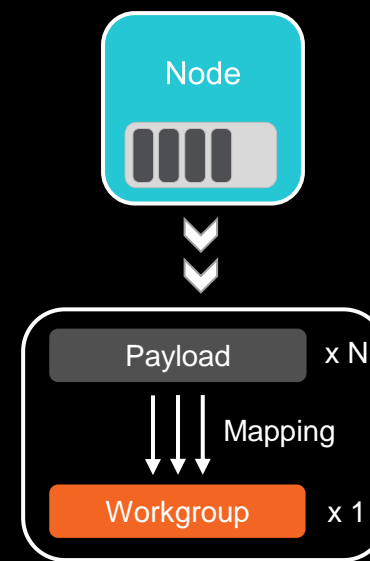
Broadcasting

Launch workgroups with fixed or dynamic size



Coalescing

Launch a workgroup for (up to) N payloads



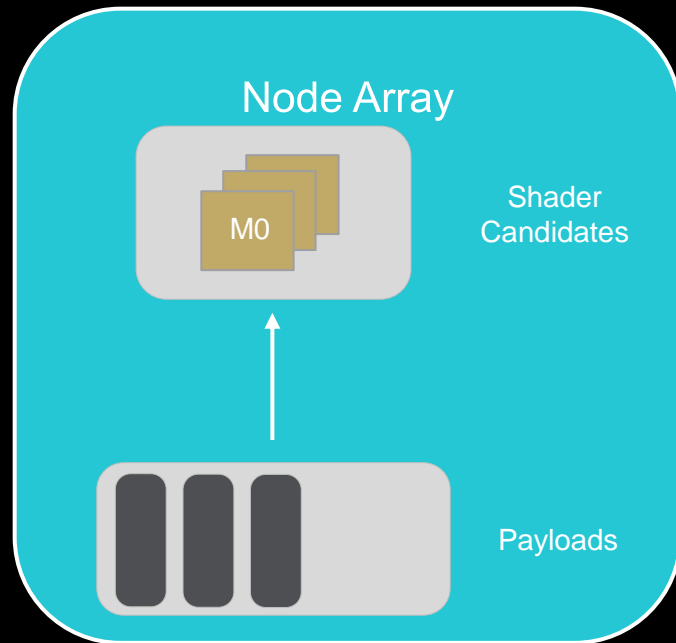
Thread

Launch one thread per payload

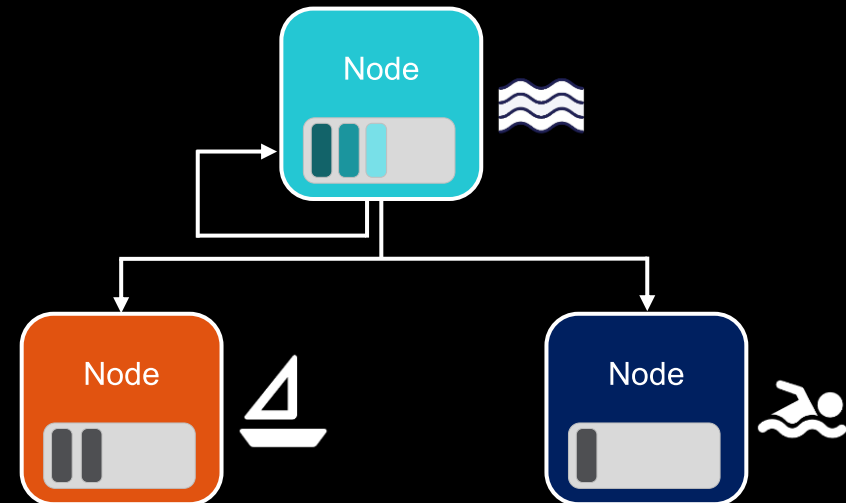
VK_AMD_X_shader_enqueue

In a nutshell

- Nodes can be node arrays
 - Shaders with the same input and launch type
 - Payloads are allocated and queued for a specific Node ID
 - Material shading



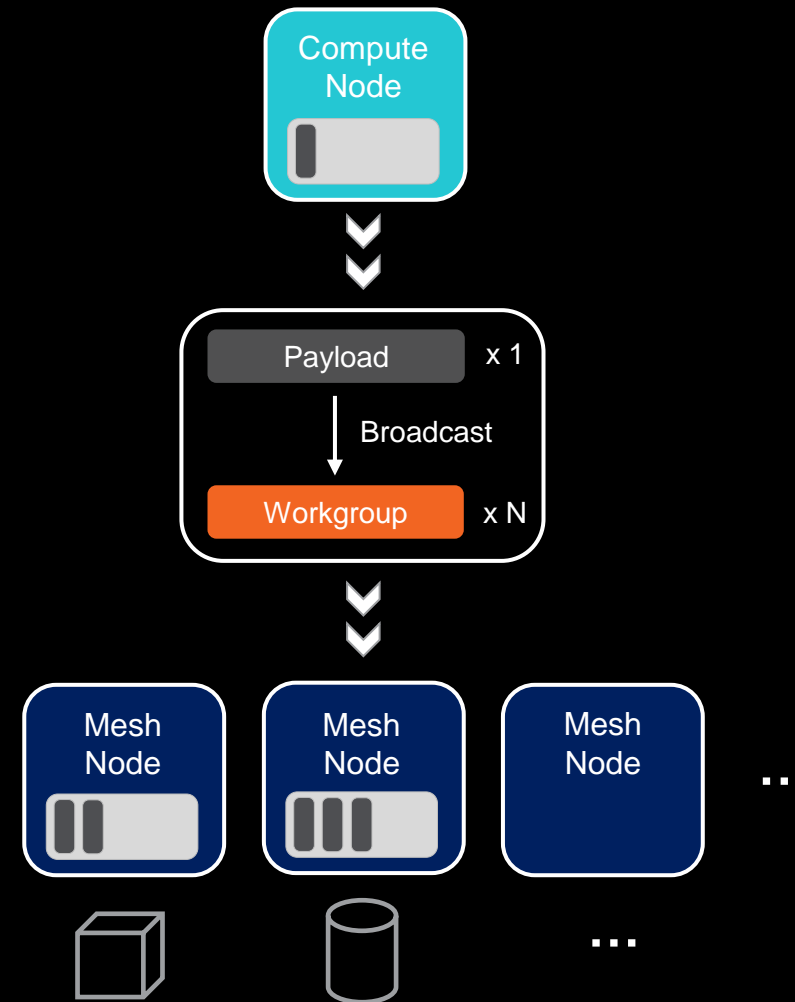
- Topology
 - Only self-recursion is allowed
 - Merge & branching are also allowed
 - Scene traversal
 - Content dependency representation



VK_AMD_X_shader_enqueue

In a nutshell

- Indirect pipeline/shader switching
 - Node replacement can be perceived as shader switching
- Mesh nodes
 - Mesh shaders that accept payloads
 - Leaf node only
 - No task shaders (replaced by compute ones)
- Memory consumption
 - No worst-case allocation
 - A range of buffer sizes is acceptable



Summary

	VK_EXT_device_generated_commands	VK_AMD_X_shader_enqueue
Memory	Worst-case sized buffer upfront	Optimized buffer allocation
Performance	Interventions from the host (e.g., barriers, address ...)	Autonomous
Flexibility	Linear	Topological graph (e.g., branch, merge, recursion ...)
Expressiveness	Customize commands layout Local representation	Producer/Consumer networks Global representation

Miscellaneous

- What to expect next?
- VK_EXT_device_generated_commands
 - Proposal: https://github.com/KhronosGroup/Vulkan-Docs/blob/main/proposals/VK_EXT_device_generated_commands.adoc
 - Registry: https://registry.khronos.org/vulkan/specs/1.3-extensions/man/html/VK_EXT_device_generated_commands.html
- VK_AMD_shader_enqueue
 - Announcement: <https://gpuopen.com/gpu-work-graphs-in-vulkan/> & <https://gpuopen.com/learn/gpu-workgraphs-mesh-nodes-vulkan/>
 - Proposal: https://github.com/KhronosGroup/Vulkan-Docs/blob/main/proposals/VK_AMD_shader_enqueue.adoc
 - Registry: https://registry.khronos.org/vulkan/specs/1.3-extensions/man/html/VK_AMD_shader_enqueue.html
 - Sample App: <https://github.com/GPUOpen-LibrariesAndSDKs/Vulkan-Samples>
 - Beta Driver: <https://www.amd.com/en/resources/support-articles/release-notes/RN-RAD-WIN-24-10-30-02.html#>
- Acknowledgement
 - Liu Mengyang, Jesionowski Maciej, Alnasser Mais, Hector Tobias, Sines Gabor
- Contact
 - shanyi.zhang@amd.com

Vulkan 1.4

DISCLAIMER

The information presented in this document is for informational purposes only and may contain technical inaccuracies, omissions, and typographical errors. The information contained herein is subject to change and may be rendered inaccurate for many reasons, including but not limited to product and roadmap changes, component and motherboard version changes, new model and/or product releases, product differences between differing manufacturers, software changes, BIOS flashes, firmware upgrades, or the like. Any computer system has risks of security vulnerabilities that cannot be completely prevented or mitigated. AMD assumes no obligation to update or otherwise correct or revise this information. However, AMD reserves the right to revise this information and to make changes from time to time to the content hereof without obligation of AMD to notify any person of such revisions or changes.

THIS INFORMATION IS PROVIDED 'AS IS.' AMD MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND ASSUMES NO RESPONSIBILITY FOR ANY INACCURACIES, ERRORS, OR OMISSIONS THAT MAY APPEAR IN THIS INFORMATION. AMD SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL AMD BE LIABLE TO ANY PERSON FOR ANY RELIANCE, DIRECT, INDIRECT, SPECIAL, OR OTHER CONSEQUENTIAL DAMAGES ARISING FROM THE USE OF ANY INFORMATION CONTAINED HEREIN, EVEN IF AMD IS EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

© 2024 Advanced Micro Devices, Inc. All rights reserved. AMD, the AMD Arrow logo, Radeon, RDNA, Ryzen, and combinations thereof are trademarks of Advanced Micro Devices, Inc. Other product names used in this publication are for identification purposes only and may be trademarks of their respective owners. Vulkan and the Vulkan logo are registered trademarks of the Khronos Group Inc.

