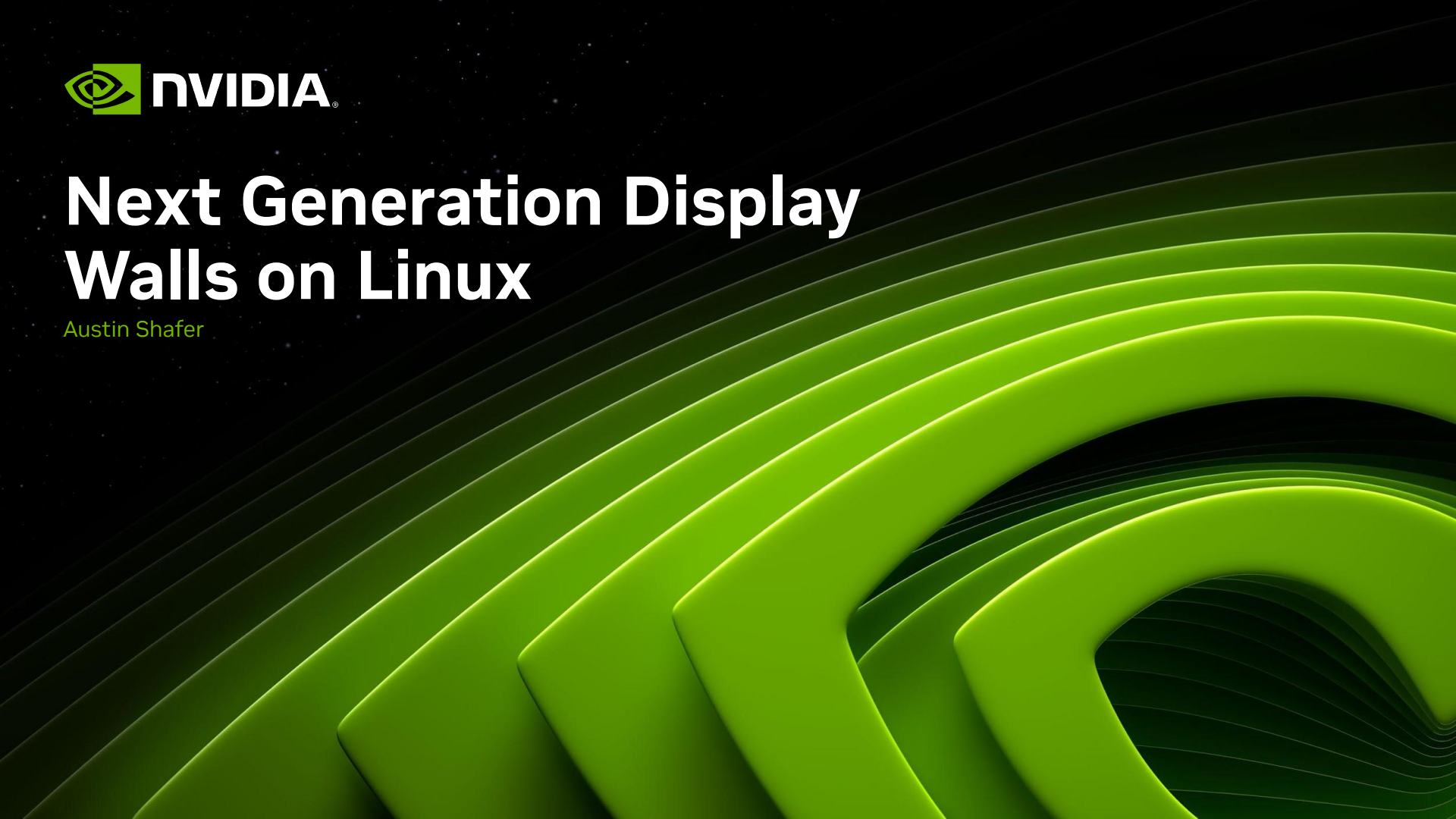


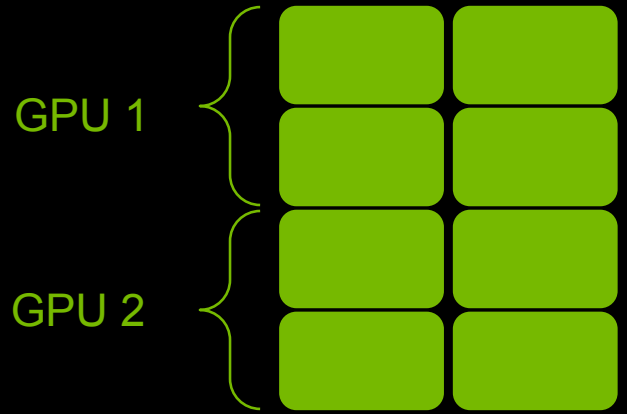
Next Generation Display Walls on Linux

Austin Shafer

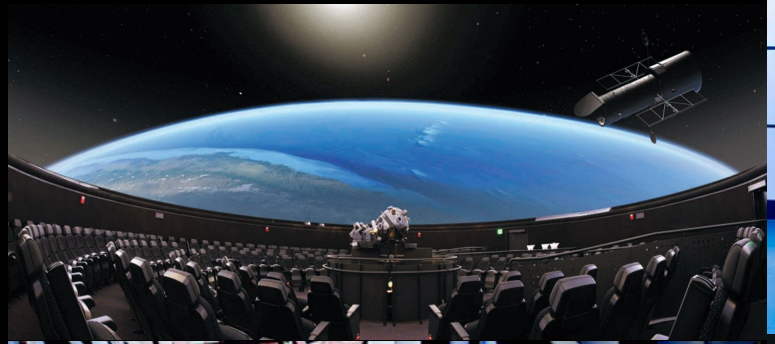
What is a Display Wall?

Also known as Video Walls, Media Walls, LED walls

- Array of displays/projectors arranged to look like one large display
- Driven by multiple devices
- Wide variety of use cases
  - Planetariums
  - Attractions/Amusement parks
  - Digital signage
  - Control centers
- Might be irregularly shaped or warped
- Most deployments use Windows, we would like to improve Linux usage



Examples

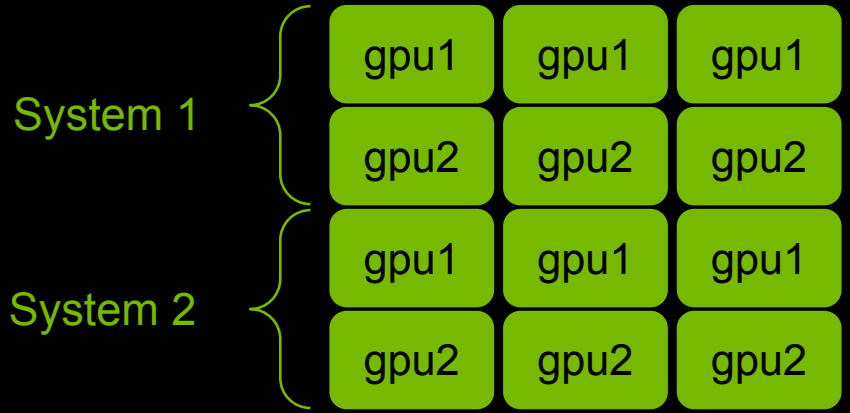


https://skyskan.com/products/digital-planetarium-systems/
https://www.electrosonic.com/projects/the-weather-channel

Display Wall Challenges

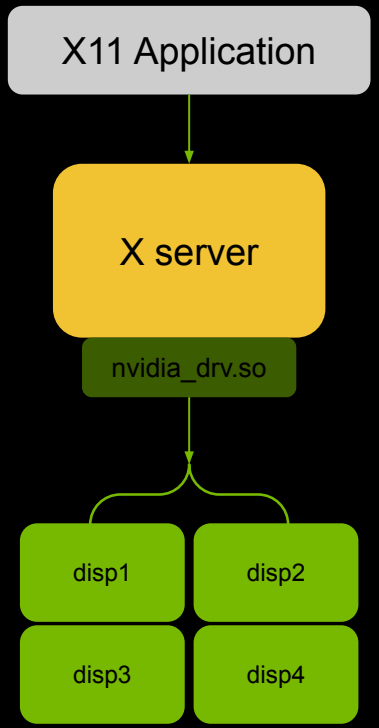
Very different from end-user systems

- Multi-GPU or even Multi-system
- Synchronized presentation
  - Hardware unit provides a synchronization signal to the systems/GPUs
  - NVIDIA hardware solution: RTX Pro Sync (aka Framelock, Quadro Sync)
- Unique customer setups
- Infrequently updated, old kernels
- Developers want direct control
  - Original implementations were developed before direct display was easily possible. DRM-KMS wasn't available and everything required a compositor.



Display Walls on Xorg

Historic solution for display walls



NVIDIA “Mosaic” displays:

- Combine multiple displays into one large virtual surface
- Desktop is one large X screen
- Implementation lives in NVIDIA X driver, highly Xorg-specific solution
- Configured via nvidia-settings and/or xorg.conf
- Suits what users need, but ecosystem is migrating away from X

Does this design translate to Wayland?

Short answer: not fully

Similar to Xorg design, but with additional challenges:

- Keeps Xorg tradeoffs - app doesn't directly drive displays, compositor has to own them
- Shipping an implementation becomes harder
  - No cross-compositor method to ship an implementation
  - Compositors probably don't want to be full of vendor-specific display code

This architecture wasn't what display wall developers wanted originally, so why should we keep forcing it?

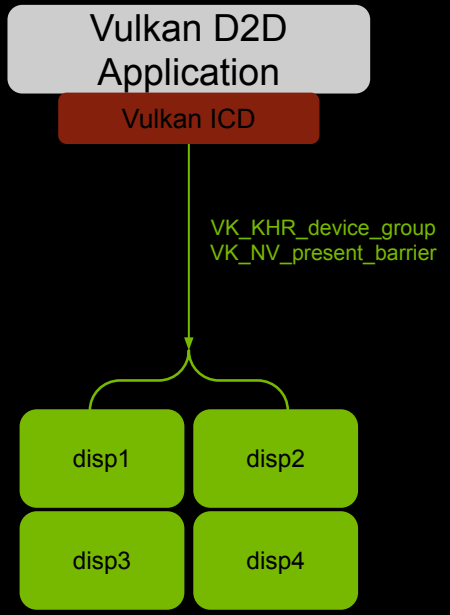
Vulkan Direct 2 Display

A wsi type which better fits what display walls want

Vulkan Direct 2 Display (VK_KHR_display) lets a vulkan swapchain drive a physical display without a window system.

Currently recommended solution, lets us:

- Direct control of displays, presentation happens in the application
- Control fine-grained multi-GPU usage, where and how copies happen
- Perform display transformations themselves
- Avoid relying on a specific window system



Vulkan D2D missing pieces

New design has new challenges

Vulkan D2D direct path works well, but still a few problems:

- Every application needs logic to handle feature calibration and configuration
  - No centralized configuration, configure each application separately
  - Applications might not support all features
- No seamless continuity between repeated runs of the application
  - Performing a modeset may take noticeable time or interrupt display features

How can we improve this? We could have a server which:

- Owns the displays
- Configures display features
- Leases those displays out

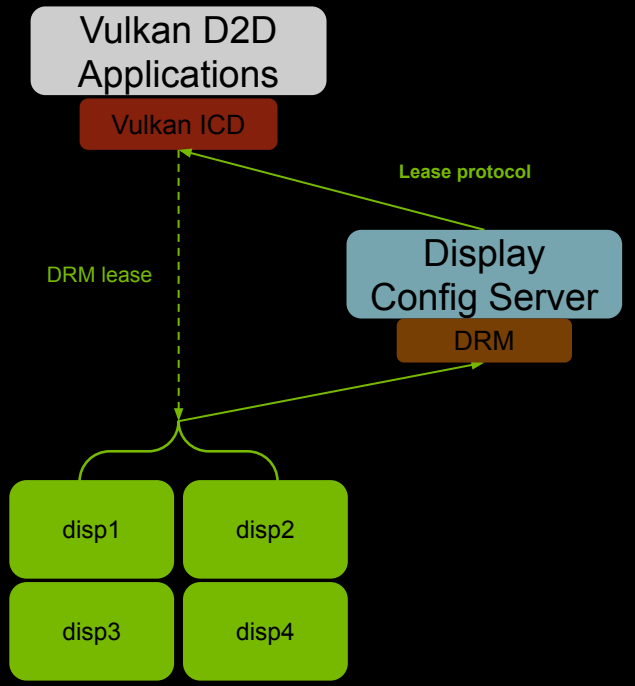
Display Config Server

https://github.com/NVIDIA/display-config-server

Display Config Server enhances the Vulkan D2D model: the server owns displays and leases them out to Vulkan clients.

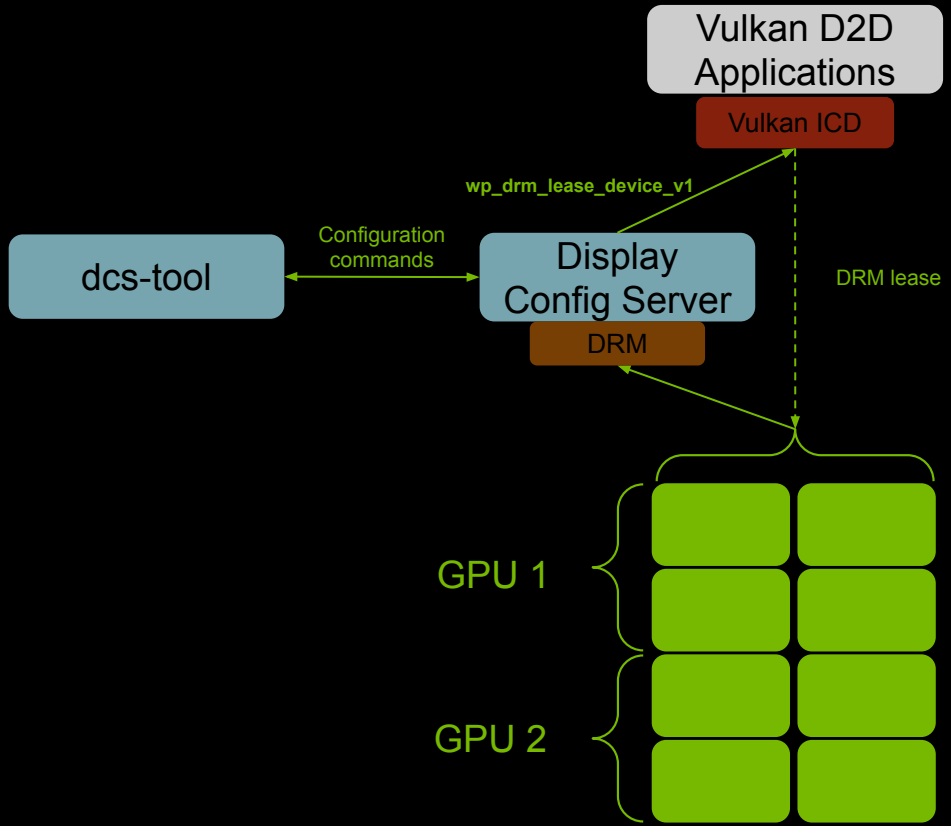
Extends current Vulkan D2D solution:

- Configuration is now centralized with the server
- Seamless handoff between Vulkan D2D applications improves experience
- Display feature enablement happens behind the scenes, app doesn't have to configure or implement anything



Linux Display Wall Example

Vulkan D2D + DCS

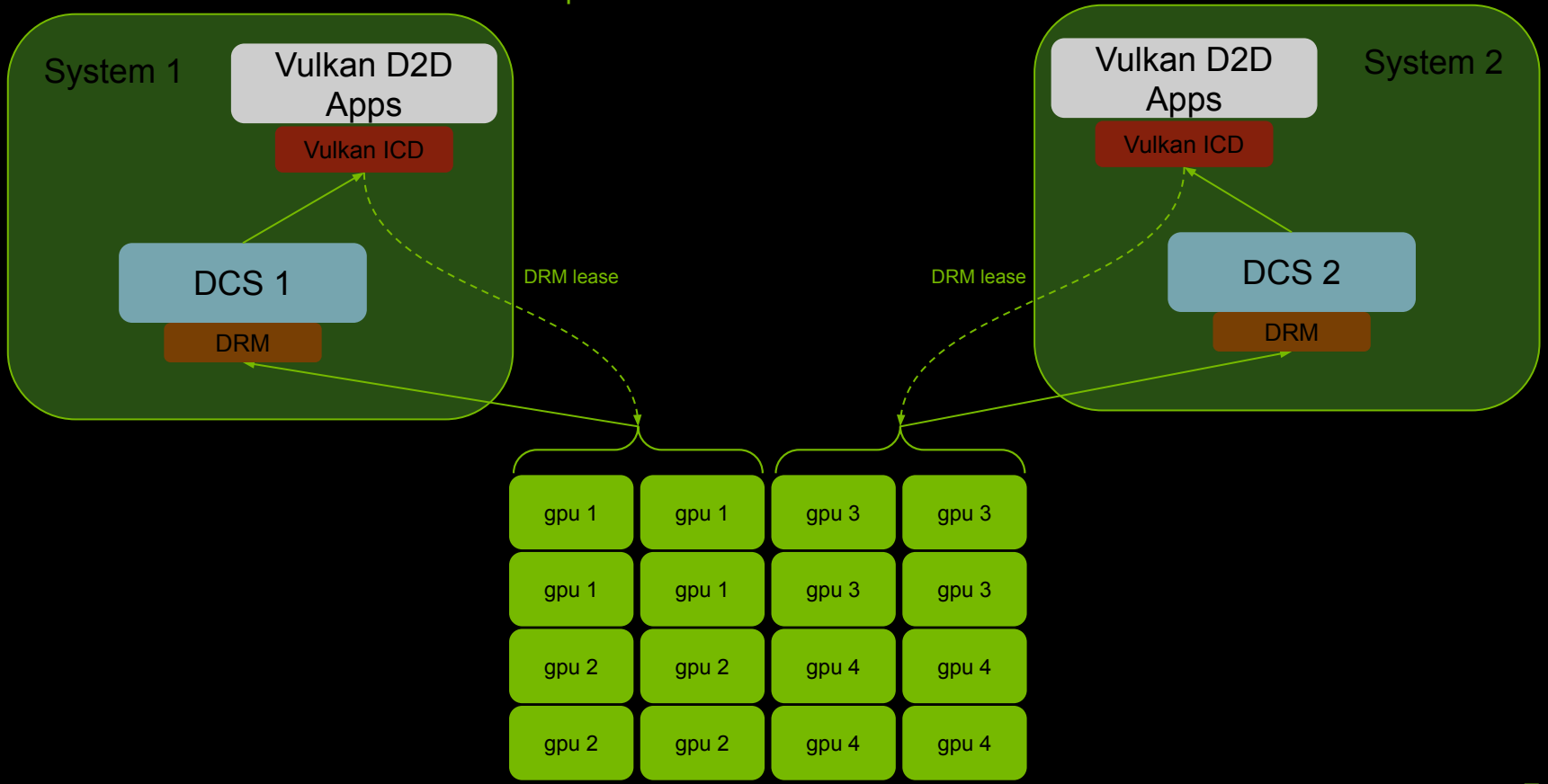


Example deployment:

- 2 GPUs, 8 displays
- Operator uses dcs-tool for configuration once
- Vulkan applications acquire displays each run
- VK_KHR_device_group drives simultaneous presentation on both GPUs

Multi-System Example

Multiple DCS and Vulkan instances



Future work

What's next?

- Support DCS running nested with a Wayland compositor
  - Compositor runs on one monitor, leases the remaining displays out
  - DCS acquires the displays, configures them, and leases them out to applications
  - This makes operating the system a bit easier
- We are working on improving how the Vulkan D2D API can present on display walls
- Goal is to help improve Linux adoption on display walls


NVIDIA