COLLABORA

Mainline Explicit Fencing

A new era for graphics

Gustavo Padovan

Open First

COLLABORA

Agenda

- Intro to Fencing
- Implicit Fencing
- Explicit Fencing
- Android Sync Framework
- Mainline Explicit Fencing
- struct fence
- Sync de-stage
- fence_array
- DRM
- Mesa
- Current Status

COLLABORA

Fencing

- Ensure ordering between operations
- Synchronize buffer sharing
  - e.g.: Between GPU and Display drivers
- Implicit fencing: userspace not aware
- Explicit fencing: userspace aware

COLLABORA

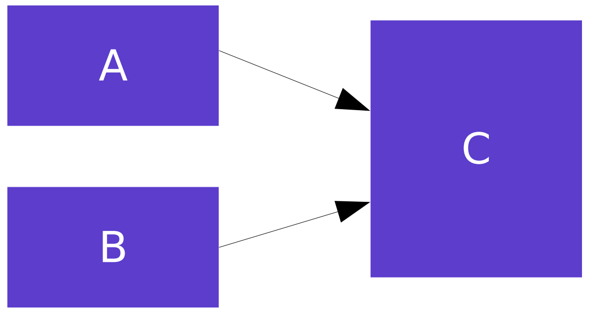
Fences

- Promise from the kernel
- Work has been queued
- Signal when finished
- Userspace and drivers wait for the signal

Implicit Fencing

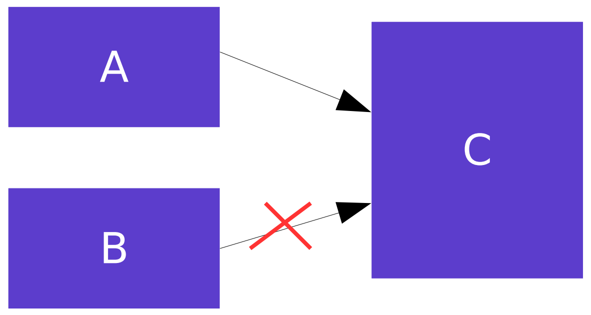
- No userspace knowledge/interference
- Simple/Dumb compositors
  - No buffer state information
- But it can freeze the whole desktop!

Implicit Fencing



- Buffer C will be composed of A and B
- Buffers A and B can render in Parallel
- Compositor notified only when both finishes

Implicit Fencing



- A is fast and B takes too long
- C is blocked waiting for both to render
- The entire desktop freezes!

Explicit Fencing

- Fences goes to userspace
- Userspace can control synchronization
- Smart decisions on compositors
- Avoid blocking the entire desktop

Explicit Fencing

- No need to wait/block in userspace
- Better for traceability/debuggability
- Vulkan requires it
  - Part of the API
  - Efficient Sub-buffer processing

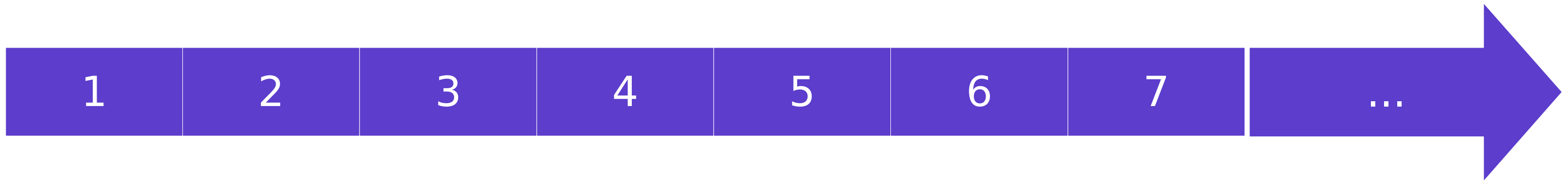
COLLABORA

Android Sync Framework

- Android Explicit Fencing implementation
- Use fd for fence passing
- Consumer-Producer queue
- **Sync Timeline** to control ordering
- **Sync Point** to represent a fence
- **Sync Fence** for fd passing

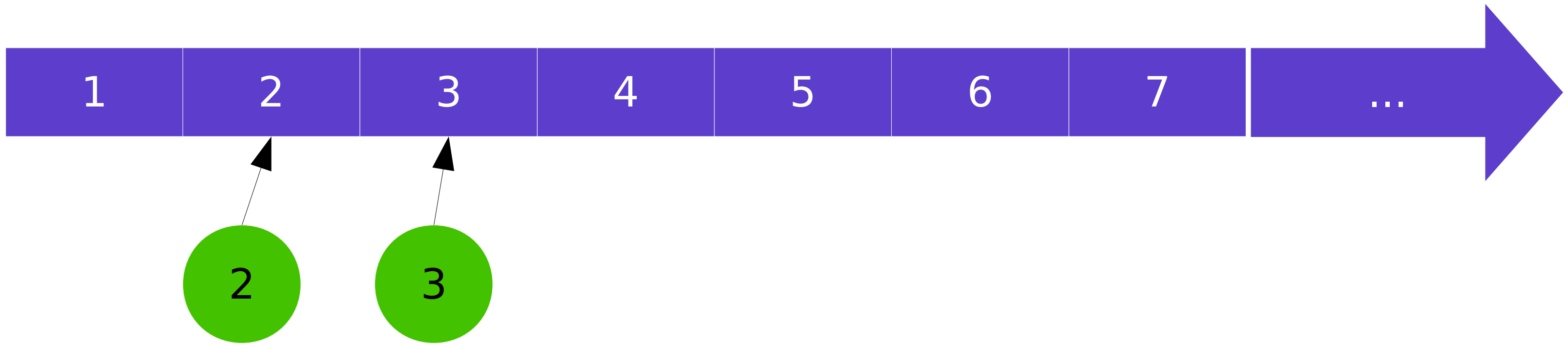
COLLABORA

Sync Timeline



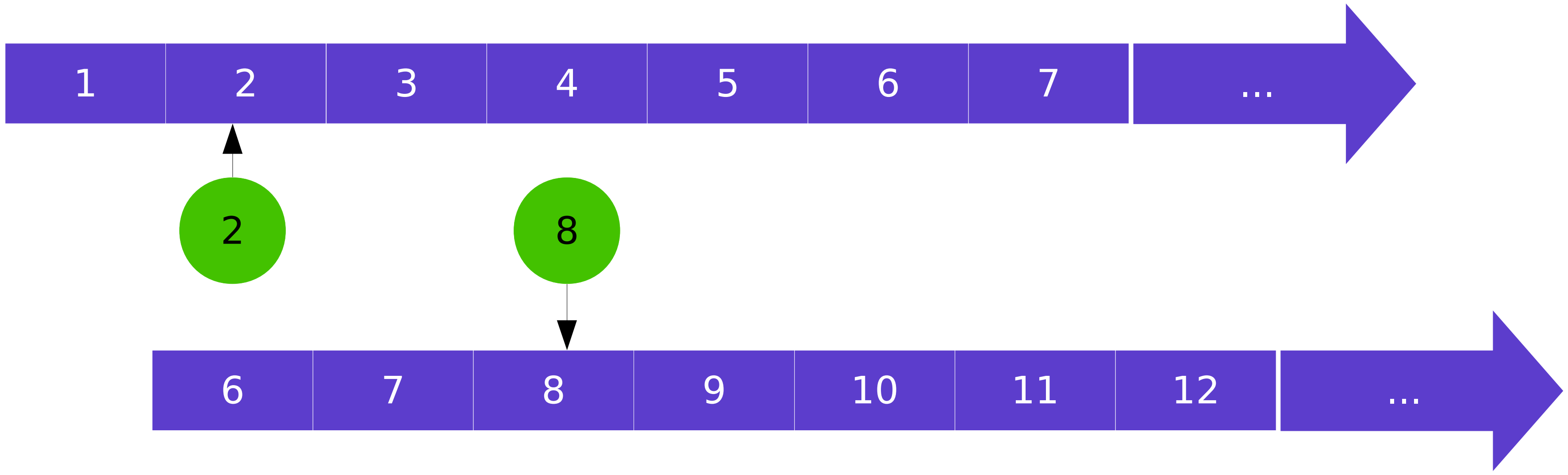
- Monotonically increasing counter
- Usually one timeline per driver context

Sync Point



- It is the fence
- Represents a value on the timeline
- Three states: active, signaled and error

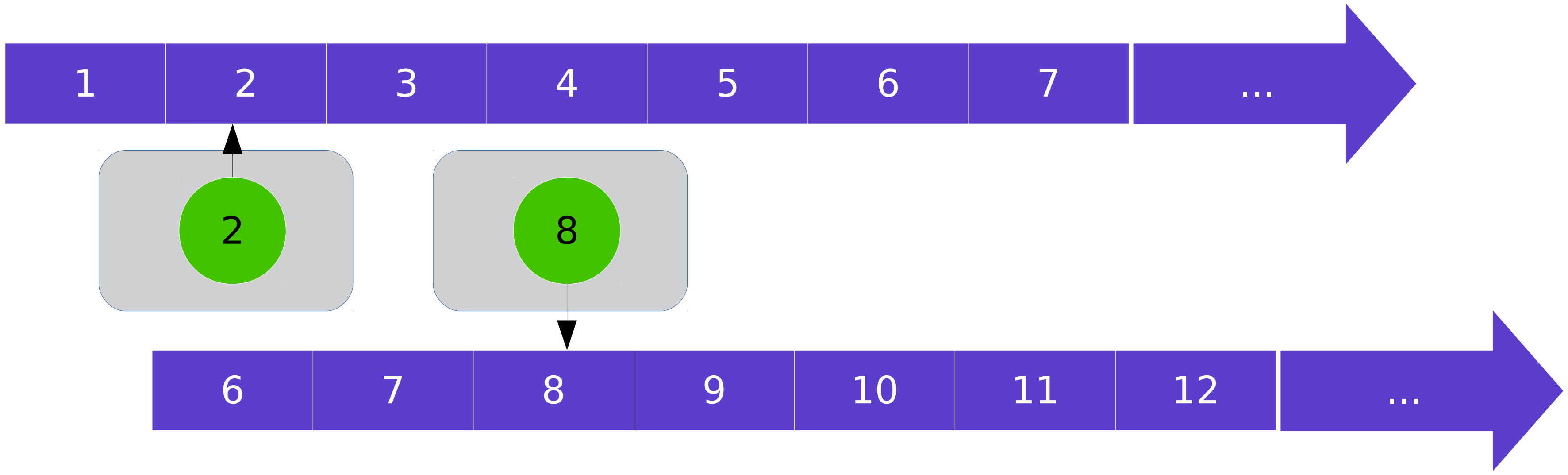
Sync Point



- Multiple timelines!

COLLABORA

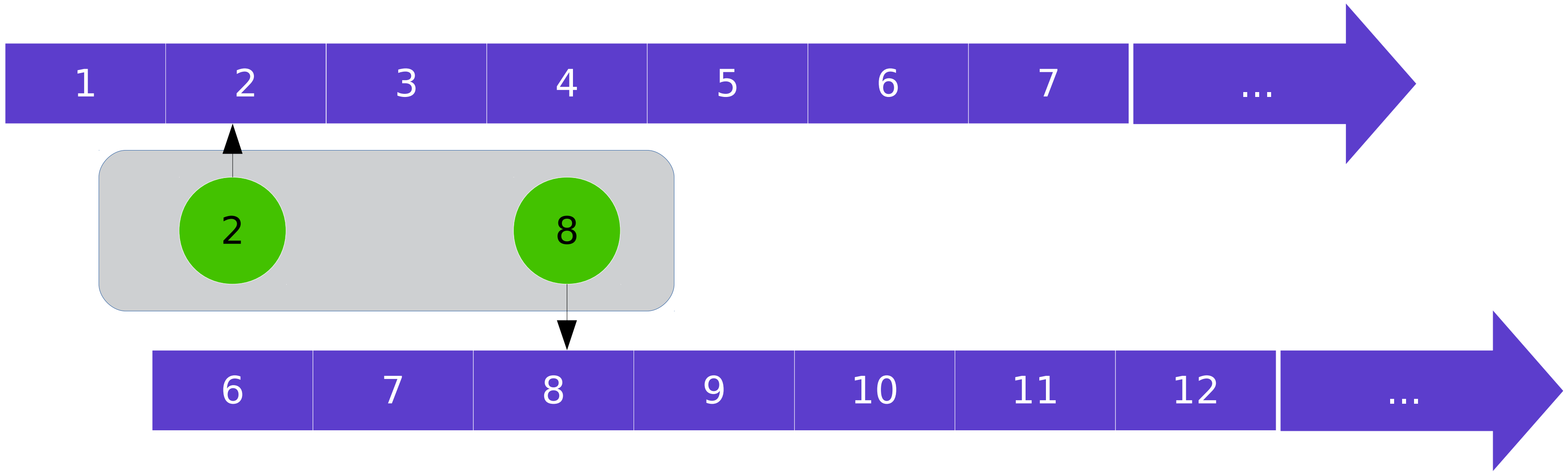
Sync Fence



- Wrap Sync Point into a file
- Also have active and signaled states
- Shared via fd-passing to/from userspace

COLLABORA

Sync Fence



- Sync fence can be merged!
- It can contain many Sync Points

COLLABORA

Android Sync Framework - ioctls

- sync_wait(fd)
- sync_merge(fd1, fd2)
- sync_fence_info(fd)

Mainline Explicit Fencing

- Started with the fence synchronization mechanism by Maarten Lankhorst
- Buffer synchronization between drivers

struct fence

- struct fence
- fence->context
- fence_signal()
- fence_wait()
- fence_add_callback()

Sync Framework de-staging

- Add Android Sync to staging in 2013
- Mainly need for fd-passing
- Removed Sync Timeline
- Removed Sync Point
- Reworked Sync Fence

COLLABORA

Sync File

- Renamed Sync Fence to Sync File
- Changed ioctl API
  - Provided patch to Android's libsync
- Removed internal kernel API
- Used strictly for fd-passing
  - sync_file = sync_file_create(fence)
  - fence = sync_file_get_fence(fd)

COLLABORA

fence_array

- Subclass of struct fence
- Store multiple fences
- Useful for merged Sync File
- Hide complexity from the drivers

DRM/KMS

- Only available for Atomic Modesetting
- Receives fences from userspace
- Wait for fence signal before scanout
- Create new fences to return buffer to pipeline
- Signal created fences at scanout
  - It means **previous** buffer can be reused
- Entirely in DRM Core

COLLABORA

DRM/KMS: in-fences

- in-fences: fences received from userspace
- FENCE_FD property on each DRM Plane
- Receives sync_file fds carrying fences
- drm_atomic_helper_wait_for_fences() helper

COLLABORA

DRM/KMS: out-fences

- out-fences: fences sent to userspace
- One fence per DRM CRTC
- Extended the DRM Atomic ioctl args
- Userspace need to ask for out-fence
  - DRM_MODE_ATOMIC_OUT_FENCE flag
  - libdrm: drmModeAtomicAddOutFences()
- get_unused_fd() + sync_file_create() + fd_install()

DRM/renderer

- Similar to KMS side
- Extends execbuffer ioctl args on each driver
- Every driver needs sync_file/fences support
- WIP on freedreno, i915 and virgl

Mesa

- EGL_ANDROID_native_fence_sync
  - Create Android fence fd
- EGL_ANDROID_wait_sync
  - Make the GPU wait for fence to signal
- WIP by Rob Clark

COLLABORA

Current Status Summary

- Sync File syncronization de-stage: DONE
- SW_SYNC validation de-stage: DONE
- fence_array: DONE
- DRM/KMS: WIP
- DRM/renderer: WIP
- MESA: WIP
- intel-gpu-tests: WIP
- Wayland: TODO

Thank you to everyone involved

Daniel Vetter, Rob Clark, Greg KH, Daniel Stone, Robert Foss, Sean Paul, Stéphane Marchesin, Maarten Lankhorst, Chris Wilson, Christian König and others.

COLLABORA

Thank you!

Gustavo Padovan
gustavo@padovan.org
www.padovan.org
www.collabora.com