A Look Inside Mutter / GNOME Shell

Georges Stavracas
Endless OS Foundation
Hi
Mutter
What’s Mutter

- Based on Clutter
  - Mutter = Metacity + Clutter
- A library to write compositors
- A tiny compositor
- Supports plugins
  - GNOME Shell is a Mutter plugin
- X11 / Wayland
Mutter, Clutter, Cogl, What?

Mutter
(Compositor)
Mutter, Clutter, Cogl, What?

Clients → libwayland-server → Mutter (Compositor)
Mutter, Clutter, Cogl, What?

Clients

libwayland-server

Mutter (Compositor)

Mesa
Mutter, Clutter, Cogl, What?

Clients → libwayland-server → Mutter (Compositor) → Mesa

Kernel
Hardware
Mutter, Clutter, Cogl, What?

Clients

libwayland-server

Mutter (Compositor)

Clutter

Cogl

Mesa

Kernel Hardware
Mutter, Clutter, Cogl, What?

Clients

libwayland-server

Mutter (Composer)

Clutter

Cogl

Mesa

Mutter (GBM / KMS)

Kernel Hardware
Mutter, Clutter, Cogl, What?

- Clients
  - libwayland-server
  - libinput

- Mutter (Compositor)
  - Clutter
  - Cogl
  - Mesa
  - Mutter (GBM / KMS)

- Kernel Hardware
On X11...

- Kernel
- Hardware
- Clients
- Clutter
- Cogl
- Mutter (WM)
- X server
Cogl

An abstraction layer over OpenGL

- Allows dealing with GL in an object-oriented manner
- Predates Vulkan by many years, yet surprisingly similar!
- Was an independent project before being merged with Mutter
- Most APIs revolve around CoglContext and CoglFramebuffer
Clutter
Clutter

An OpenGL-based toolkit

- Previously, and application toolkit
- Forked, copy lives inside Mutter
- Slowly progressing to be a compositor toolkit
- Actors, stage, views, etc, allow creating nice effects
- Animation framework
- Input handling
Clutter

- Traditional toolkit update cycle:
  - Layout: places actors somewhere
  - Paint: paint actor contents
  - Pick: determine what's beneath the cursor
- 2D actor tree on 3D space
- Uses paint volumes for basic culling
- Affine transformations everywhere
Frame Clock
Frame Scheduler

- Recently the frame scheduler was improved
- Goal was to give more room for applications to draw
- ... and higher chances for reducing latency
Frame Scheduler

| → monitor vsync
| → next frame
| → skip frame
Frame Scheduler (improved)

| → monitor vsync
| → next frame
| → skip frame
Transactional KMS

... or, at least, the path to
Non-transactional (until GNOME 3.32)

 GNOME Shell

Kernel

Author: Jonas Ådahl, Red Hat
Fake Transactional (current)

 GNOME Shell

 Transaction

Kernel

Frame

Configure #1
Configure #2
Configure #3

Command #1
Command #2
Command #3

Frame

Author: Jonas Ådahl, Red Hat
Atomic Transactional (TBD)
Transactional KMS

- Updates KMS state and composited image atomically
- Needs to be able to switch between OpenGL and:
  - Hardware overlays for compositing
  - Accelerated cursor plane
  - CRTC gamma for color management
- DRM buffer modifiers
- Most of the code lives under src/backends/native/meta-kms-*
- Thread safety in mind
  - Eventually, a dedicated KMS thread
  - Multi-threaded KMS transaction setup
While Talking About KMS...

- Basically no support for overlay planes yet
  - Mutter uses the cursor plane at most
- Will require deep surgery in Mutter’s Clutter
- Use libliftoff perhaps?
  - Seems like the most reasonable option for now
Input
Input

- Uses libinput and xcb
- Implemented as seats
  - Device owners and source of events
  - Wayland: MetaSeatNative
  - X11: MetaSeatX11
- Next steps: input thread
  - Work in progress:
    https://gitlab.gnome.org/GNOME/mutter/-/merge_requests/1403
Input Thread

Input

UI Wayland

KMS

Author: Carlos Garnacho, Red Hat
Input Thread

Input

UI Wayland

KMS

Author: Carlos Garnacho, Red Hat
Potential Benefits

- No missed libinput events
- No blocked cursor pointer
- Better handling of high frequency devices
- Reuse of cursor plane (e.g. for tablets)
- In general, peace of mind
Screen Sharing
Screen Sharing

- Based on PipeWire
- Pre-allocates a number of buffers
- Cycles through them
- Supports streaming metadata
- Fast path: DMA-BUF sharing
- Slow path: `glGetError()`
Screen Sharing (fallback)
Screen Sharing (DMA-BUF)
Screen Sharing

- Demo!
Screen Sharing

- No NVidia support
- No X11 support

Some thoughts:

- Dealing with GBM is a breeze
- PipeWire doesn’t send critical DMA-BUF info, such as modifiers
  - This shouldn’t be hard to add to PipeWire though
  - Great task to start contributing
Screen Sharing

- Links
  - https://github.com/obsproject/obs-studio/pull/3338
  - https://gitlab.gnome.org/feaneron/obs-xdg-portal/-/merge_requests/5
- If you have a modern Linux distribution:

  ```bash
  $ flatpak install --user https://flathub.org/beta-repo/appstream/com.obsproject.Studio.flatpakref
  ```
Other Plans
Other Plans

- Finish transition to Graphene
- Retained rendering tree
- Deeper, more detailed profiling
Graphene

- A graphics data types library
- Platform-specific optimizations: SSE 2 & 4.1, ARM NEON, etc
- Jackpot: graphene matrices!
- Mutter is halfway through the transition
  - Finish porting to Graphene
  - Only basic data types ported
  - Last battle is CoglMatrix → graphene_matrix_t
  - Help needed
Graphene
Q&A