Waylandifying Chromium - From Downstream To Shipping

Maksim Sisov - msisov@igalia.com
• About Igalia.
• History of the Project.
• What is Ozone?
• How Ozone Integrates Inside Chromium.
• Ozone/Wayland Limitations:
  ○ Limitation 1 - Graphics Pipeline.
  ○ Limitation 2 - Tab Dragging Feature.
• Performance of Chromium on Wayland:
  ○ Performance Comparison - non-Ozone/X11 vs Ozone/Wayland.
• Shipping of Ozone/Wayland.
• Bonus: Migration of X11 to Ozone.
About Igalia

- Worker-owned; employee-run Open Source consultancy company; based in Galicia, Spain.
About Igalia

- ~90 employees around the world.
- Areas of expertise:
  - **Browsers and Client-side Web Technologies**
    - Chromium/Blink, WebKit (WPE & WebKitGTK+), Firefox and Servo
  - **Graphics Pipeline and Rendering Technologies**
    - Hardware accelerated 3D APIs, Mesa open source OpenGL (ES), Vulkan drivers, and more
  - **Compilers and Programming languages**
    - JavaScript Contributor through TC39
    - V8, JavaScriptCore, ChakraCore, SpiderMonkey
  - **Multimedia**
    - GStreamer, VA-API, MediaSDK, and more
  - **Embedded Linux and Device Drivers**
    - From Graphics to Networking
  - **Accessibility Tools and Technologies**
    - Assistive Technologies in Chrome, Firefox or Safari
  - **Virtualization and Cloud**
    - QEMU/KVM, CEPH
About Igalia

- **World Wide Web Consortium (W3C)**
  - *Co-chairs the W3C’s ARIA working group*
- **Linux Foundation**
  - *Also Member of Automotive Grade Linux Steering Committee*
- **Khronos Group**
  - *Participates actively in the Khronos Conformance Testing Working Group focusing on the OpenGL, OpenGL ES, and Vulkan APIs*
- **ECMA**
  - *Chairs the ECMA JavaScript TC39 group*
- **WHATWG**
- **GENIVI® Alliance**
- **Software Freedom Conservancy (SFC), Electronic Frontier Foundation (EFF), AGASOL**
About Igalia
History of the Project
History of the Project

- **Started in May, 2016**
  - Intel’s Wayland implementation was not suitable for upstream.
- **Was initially based on Mus+Ash project**
  - UI as a service.
- **Moved to downstream in March, 2017.**
  - Unclear future of the Mus+Ash project.
- **Was finally decided to use a direct Aura integration in April, 2018.**
  - **Aura** - Chromium’s own windowing system
  - Moved back to upstream
- **Mus+Ash was finally discontinued in 2019.**
  - Not really related to the current project, but fact.
- **Ozone is part of regular Chrome/Chromium builds since September, 2020.**
  - Ozone can be enabled with “--enable-features=UseOzonePlatform --ozone-platform=wayland/x11” runtime flag.
- **Ozone is default on Linux - H2/2021 (hopefully).**
  - Non-Ozone/X11 (legacy X11) path is removed.
What is Ozone?
What is Ozone?

- Abstraction layer beneath the Aura window system.
  - Aura - Chromium’s window system

- Goal - make porting of Chromium to other platforms easy:
  - Interfaces, not ifdefs:
    - #if defined(USE_X11)
  - Flexible interfaces:
    - No overly prescriptive interfaces.
  - Runtime binding of platforms:
    - One binary == many backends.
  - Easy out-of-tree platforms:
    - Support for different backends.
How Ozone Integrates Inside Chromium
Aura/Ozone Design
Aura/Ozone Interaction
Browser Process

WindowTreeHostPlatform

- void ShowImpl();
- void HideImpl();
- void SetBoundsInPixels(...);
- void SetCapture();
- ...
- std::unique_ptr<PlatformWindow> window_

OzonePlatform

- OzonePlatform* GetInstance();
- void InitializeForUI(Properties);
- void InitializeForGPU(Properties);
- std::unique_ptr<PlatformWindow> CreatePlatformWindow(Properties);
- std::unique_ptr<InputMethod> CreateInputMethod();
- ...
- ...
- std::unique_ptr<InputMethod>
Design of Ozone/Wayland
Ozone/Wayland Limitations
Ozone/Wayland Limitations

- **Wayland EGL**
  - Required access to WaylandConnection.
  - Could only be used with “in-process-gpu”.

- **Tab Dragging Feature**
  - Required additional extension.
  - Wasn’t possible to use original Drag&Drop extension because of some Chromium’s internal assumptions.
Limitation 1 - Graphics Pipeline
Graphics Design

● Consists of GPU and Browser process side objects
  ○ WaylandBufferManagerHost
  ○ WaylandBufferManagerGpu
● Uses surfaceless drawing
  ○ SurfacelessEGL
● Uses Mojo for communication
  ○ Associated pipe that ensures order of messages
● Uses libgbm to create dmabuf
  ○ Reused existing implementation from Ozone/Drm
● Allows to reuse GpuMemoryBuffers framework without any modifications
  ○ Ozone/Drm is the first user
Limitation 2 -
Tab Dragging Feature
Limitation 2 - Tab Dragging Feature

● Problem
  ○ Cannot properly “swallow” and “unswallow” a surface.
  ○ Chromium uses a so-called “preview mode” for windows that are being dragged
  ○ Existing DnD protocol has limited functionality

● Solution
  ○ Implement protocol extension that would allow to reuse existing surface as a drag icon and allow a further reuse.

● More details -
  https://docs.google.com/document/d/1s6OwTi_WC-pS21WLGGQYI39yw2m42ZIVolUXBclljXB4/edit#heading=h.gidgxs
Performance Comparison - non-Ozone/X11 vs Ozone/Wayland
Raspberry Pi3 B+
Shipping of Ozone/Wayland
Shipping of Ozone/Wayland

● Mostly all the features are in place except tab drag.

● Most test suites are exercised with ozone/wayland backend:
  ○ Various browser, content browser tests, and unit tests,
  ○ More test suites are to be enabled including gpu tests,
  ○ See linux-ozone-rel bot.

● Ozone is part of Chrome releases since M87.
  ○ Can be used with "--enable-features=UseOzonePlatform
    --ozone-platform=wayland/x11" runtime flag.

● Ozone/Wayland cannot be shipped because of X11.
Bonus:
Migration of X11 to Ozone
Migration of X11 to Ozone

- Started around May/June, 2019.
- Shared most of the non-Ozone/X11 implementation with Ozone/X11
  - Including non-X11 specific bits that Ozone/Wayland could reuse
    - Like StatusIconDbus.
- Switched Ozone from compile-time to runtime switch.
  - Happened in September, 2020,
  - Can be tried with “--enable-features=UseOzonePlatform --ozone-platform=wayland/x11”.
- Can be tracked at https://crbug.com/789065
Thank you

Maksim Sisov - msisov@igalia.com
Nick Diego Yamane - nickdiego@igalia.com
Alexander Dunaev - adunaev@igalia.com
Antonio Gomes - agomes@igalia.com