From Click to Pixel: A Tour of the Linux Graphics Stack

Carl Worth
carl.d.worth@intel.com

2009-12-14

Outline

- Overview of the Linux graphics stack (2D and 3D)
- Some current changes
- The plan for the future
- Debugging performance

Alphabet Soup

Alphabet Soup

Qt Arthur GTK+ Pango Cairo

OpenGL Mesa GLX AIGLX

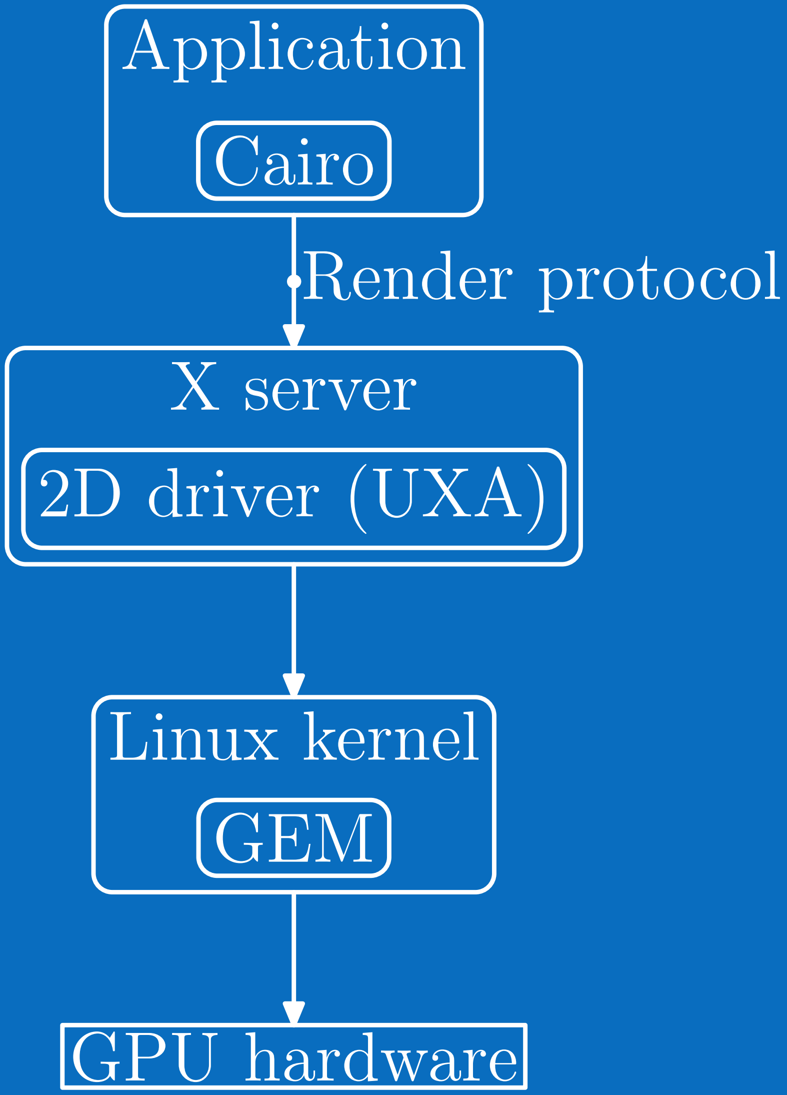
Gallium Glitz Glamor

X XAA Render EXA UXA

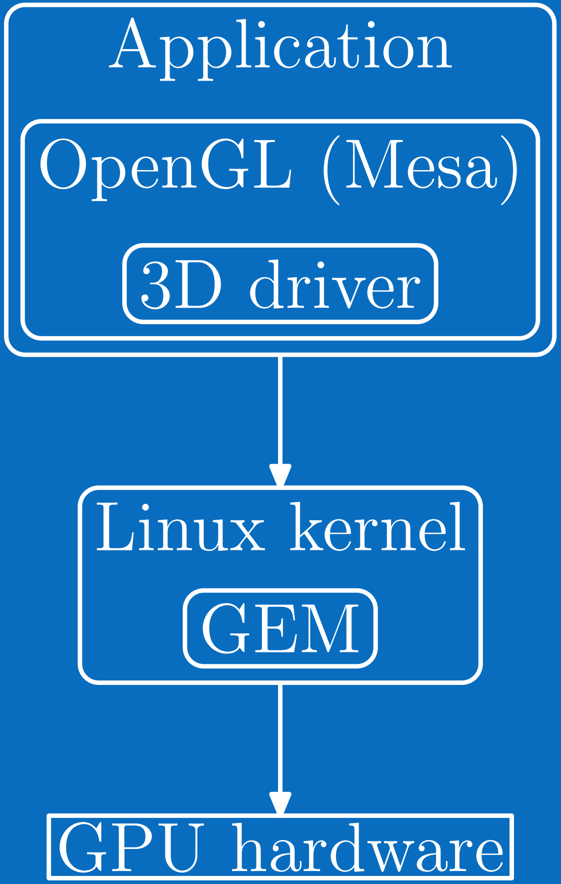
DRI DRI2 DRM TTM GEM

Stack Overview

2D Graphics Stack

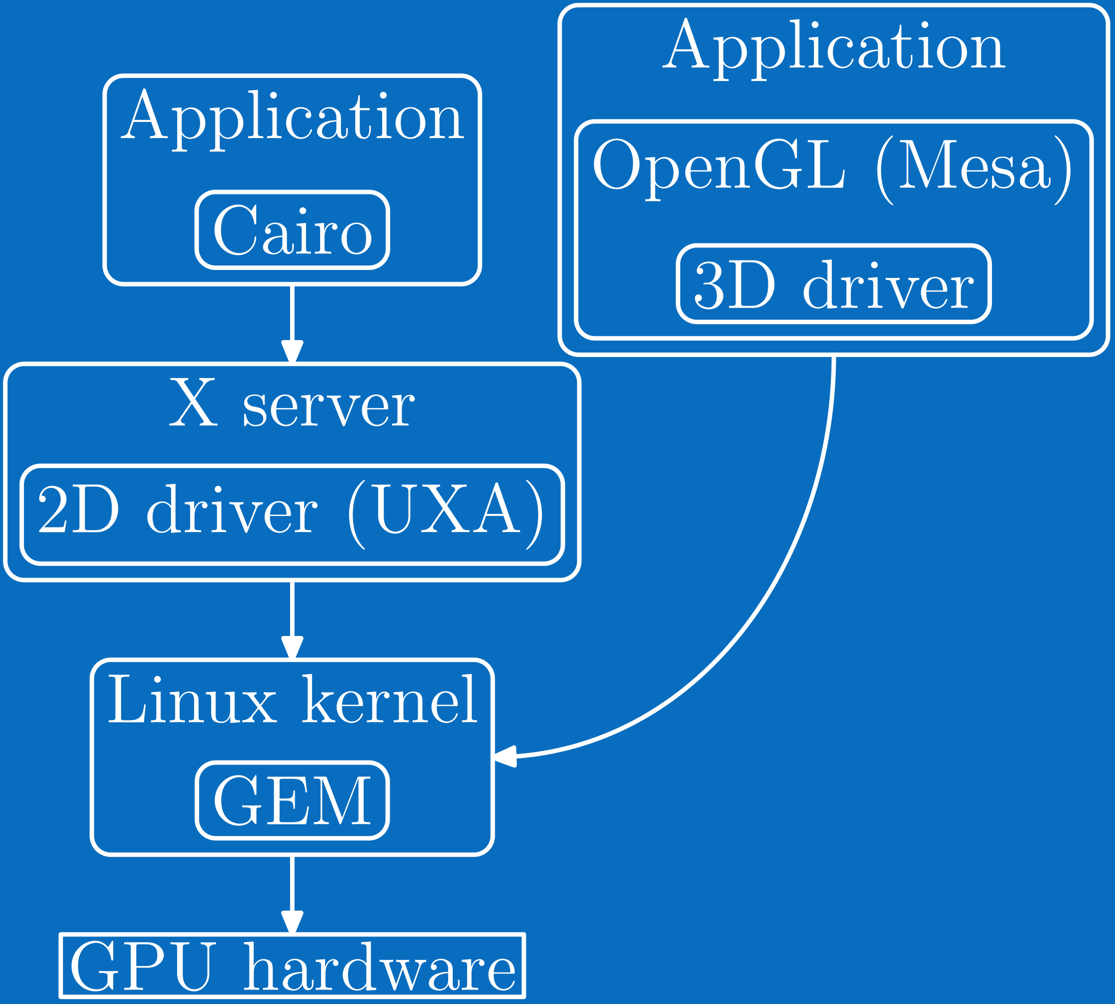


3D Graphics Stack

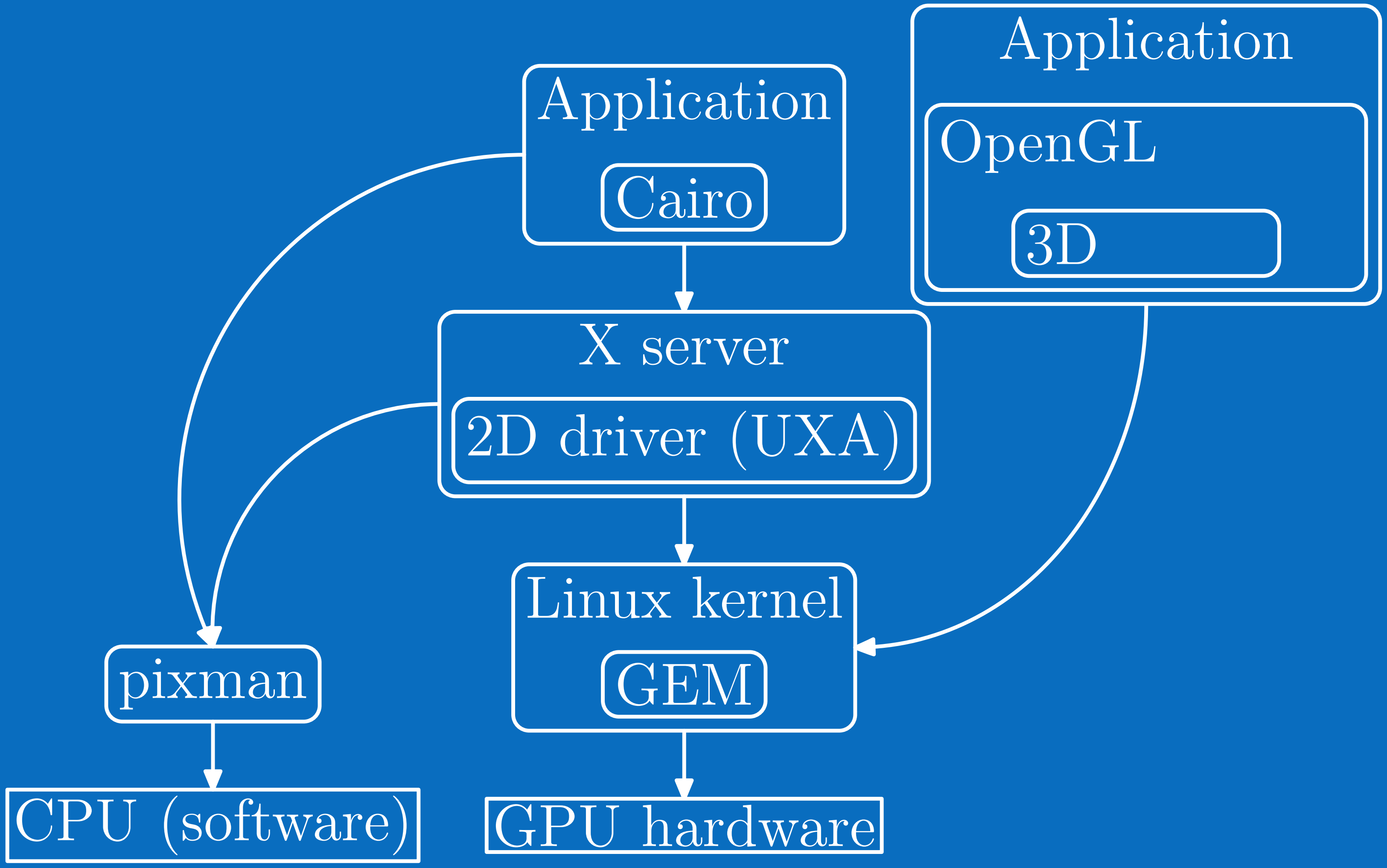


Mixing things up

Combined stack



Software fallbacks



Direct-rendering with cairo (cairo-drm)

- Chris Wilson/Kristian Høgsberg (both with Intel now)
- Functional prototypes for both 915GM and GM965
- Sets the performance target for cairo-gl
    - Gradients are 100 - 120x faster
    - Some painting operations are 50x faster
    - Text is 4x faster
- Not a viable long-term solution due to driver duplication

Everything through OpenGL

- Single driver for all future hardware
  - Optimizations to support 2D well
  - Subpixel-rendering for text
  - Glamor and cairo-gl

Stack performance

Where's the bottleneck?

- CPU Bound?
- GPU Bound?
- Just waiting

CPU Bound

- Test with top
  - Look for CPU usage of 90 - 100%
- Investigate with sysprof

GPU Bound

- Test with intel_gpu_top
  - Look for "ring idle" of 0 - 10%
- Investigate your shaders
- Useful environment variables:

INTEL_DEBUG=wm	Dump out fragment shader assembly
INTEL_DEBUG=vs	Dump out vertex shader assembly

Neither CPU nor GPU busy

Investigate with perf

Requires kernel 2.6.32 with:

- CONFIG_EVENT_TRACING=y
- CONFIG_PERF_EVENTS=y
- CONFIG_TRACING=y
- CONFIG_TRACING_SUPPORT=y

Then use perf tools:

- perf record
- perf report
- perf annotate

http://dri.freedesktop.org/wiki/IntelPerformanceTuning

Creating 2D Benchmarks (cairo trace)

- Robust capture of all (cairo-based) 2D rendering
- No modifications to application or cairo required

- HOWTO:
  - Install cairo 1.9 or later
  - cairo-trace ./my-program
  - See results in my-program.$PID.trace
  - Replay benchmark with cairo-perf-trace