A look into 2D performance
Discussion about the current architecture

Martin Peres
(Intel OTC, Finland)
Agenda

- Introduction
- Users of X-server’s 2D acceleration framework
- Evaluation of the performance
- Future for toolkits
- Discussion
Introduction

- The X11 protocol provides rendering primitives
- The primitives are rendered on the X-Server
- When GPU appeared, these primitives got accelerated
- The XRender extension extended this set of primitives
Users of the X11 2D primitives

● Toolkits:
  ○ Xt: Used by Motif and other ancient toolkits
    ■ All the rendering is done by the X Server
  ○ Gtk: Base of the Gnome Desktop Environment
    ■ Uses Cairo for all the rendering (X11 and XRender)
    ■ Cairo then uses the X Server for rendering
  ○ Qt: Base of the KDE desktop environment
    ■ Uses XRender for AA-text and cursor alpha blending
    ■ Uses the GPU or the CPU for all the other cases

● Applications:
  ○ Plenty of X demos and other legacy apps (xeyes, ...)
  ○ Applications should never use the X11 API directly
Performance evaluation

- Performance evaluated on:
  - Intel platform (Haswell GT2) on Arch Linux
  - X-Server and Intel DDX from git (September 15th)
  - Mesa from git (September 14th)
  - Power is read through RAPL (verified with a meter)

- Benchmarks:
  - Cairo demos: various demos, fixed execution time
  - Cairo traces: traces of actual 2D applications (fixed workload)
## Performance: Cairo demos

### xf86-video-intel:

<table>
<thead>
<tr>
<th>Demo name</th>
<th>Performance ratio (XLib/CPU)</th>
<th>FPS/W ratio (XLib/CPU)</th>
</tr>
</thead>
<tbody>
<tr>
<td>x11:cairo:demo:fish</td>
<td>32.6</td>
<td>15.1</td>
</tr>
<tr>
<td>x11:cairo:demo:chart</td>
<td>3.05</td>
<td>1.12</td>
</tr>
<tr>
<td>x11:cairo:demo:tiger</td>
<td>1.51</td>
<td>0.73</td>
</tr>
</tbody>
</table>

### xf86-video-modesetting:

<table>
<thead>
<tr>
<th>Demo name</th>
<th>Performance ratio (XLib/CPU)</th>
<th>FPS/W ratio (XLib/CPU)</th>
</tr>
</thead>
<tbody>
<tr>
<td>x11:cairo:demo:fish</td>
<td>31.8</td>
<td>14.9</td>
</tr>
<tr>
<td>x11:cairo:demo:chart</td>
<td>1.24</td>
<td>0.69</td>
</tr>
<tr>
<td>x11:cairo:demo:tiger</td>
<td>1.16</td>
<td>0.90</td>
</tr>
</tbody>
</table>
Performance: Cairo traces

**xf86-video-intel:** Perf Min/Avg/Max: -50/+44/+182%

2D performance of cairo traces (intel)

Comparing the xlib and cpu backends
Energy efficiency: Cairo traces

**xf86-video-intel**: Efficiency Min/Avg/Max: -73/-5/+137%

2D performance on cairo traces (intel)
Comparing the xlib and cpu backends
Performance: Cairo traces

Modesetting: Perf Min/Avg/Max: -90/-26/+123%

2D performance of cairo traces (modesetting)
Comparing the xlib and cpu backends
Energy efficiency: Cairo traces

**Modesetting:** Energy Min/Avg/Max: -97/-43/+56%

*Energy usage of cairo traces (modesetting)*

Comparing the xlib and cpu backends
Future for toolkits

- X-server is not always present anymore (Wayland, KMS)
  - Toolkits should not depend on X11 for rendering

- Qt already mostly moved away from 2D accel using X11
  - They invested in their own CPU/GPU renderer

- GTK is replacing Cairo with GTK Scene Kit (GSK)
  - GSK: Scene graph + GPU rendering
Discussion

- Rendering 2D on X is
  - barely faster than CPU rendering, unless compositing
  - using more energy than using the CPU
  - not going to work for Wayland or KMS-only apps
  - not supporting 3D transformations

- Toolkits are moving away from relying on X for rendering
  - Qt has its own CPU renderer and GPU acceleration
  - GTK is moving to [GSK](#)

- Should we tell applications to use client-side rendering?
  - CPU: Cairo, Qt
  - GPU: SKIA, Fast-UI-Draw