The Raspberry Pi Browser

Presented by
Gustavo (kov) & ChangSeok (changseok)

Brought to you by
Marco Barisione, Emanuele Aina, Julien Isorce, ChangSeok OH, Tomeu Visozo, George Kiagiadakis, Andre Magalhães, Gustavo Noronha
The Raspberry Pi

Cheap hardware, not very powerful

- ARM v6 CPU
- Up to 512 MB of RAM
- Reasonably powerful VideoCore media processor
Challenge

Make a modern browser that runs well enough to be usable

- Multiple tabs
- Responsive
- Support YouTube
Drivers and concerns

- WebKit2 out of the question for worries of memory usage overhead (working on it…)
- Accelerated Compositing out, unreliable GL stack
- Wayland postponed
Epiphany gets tab limits

- Up to 3 tabs are kept live
- Background tabs may get freed
- Per-tab session restore infrastructure reused to reload tabs that got freed when switched to
- *For the future*, could make Epiphany kill web processes, with WebKit2
Memory pressure handler

- Notification from the system which allows throwing away caches and other temporary memory when memory is becoming very scarce
- Implemented by Apple for Mac & iOS ports
- Patch on bug 123532
- Problem: only works when inside a cgroup
Disk Image Cache

- Saves decoded images to files and memory maps them
- Plugs the main source of heap fragmentation – process memory usage grows much less
- A bit different: Apple uses it for non-decoded image data
- Made cross-platform on bug 124167
- Gustavo would like to finish this patch and enable it on WebKit2GTK+, need to discuss API
Use RGB16 throughout

- This is the format cairo and pixman have been optimized for on the raspberry pi
- All surfaces changed to use it
- GStreamer backend changed to render to it directly
- Image decoders changed to decode to it directly
OMX and dispmanx usage

• OMX used for image decoding
• OMX used for video scaling
• dispmanx used directly for fullscreen video
Faster scrolling

- Tiled backing store enabled to make scrolling more responsive
- Suspends pretty much everything while scrolling – JS, animation, painting
Videos made faster

- First of all, no preloading, special case for embedded YouTube videos
- Videos are painted directly to the final surface instead of going through the tiled backing store
- YouTube hacked by injected JS to always use a `<video>` tag and to avoid running lots of its own JS code, in particular the one that deals with loading comments
Questions?

ChangSeok OH
changseok.oh@collabora.co.uk

Gustavo Noronha Silva
gustavo.noronha@collabora.co.uk