Chrome on Mobile @ 60fps

Now and in the Future

Daniel Sievers, Nat Duca

Chrome!

- As-fast-as-we-can-be web browsing on all platforms
- Since KitKat / 4.4, also Android WebView
- One codebase across all platforms

Chrome!

Chrome android just a variant on desktop

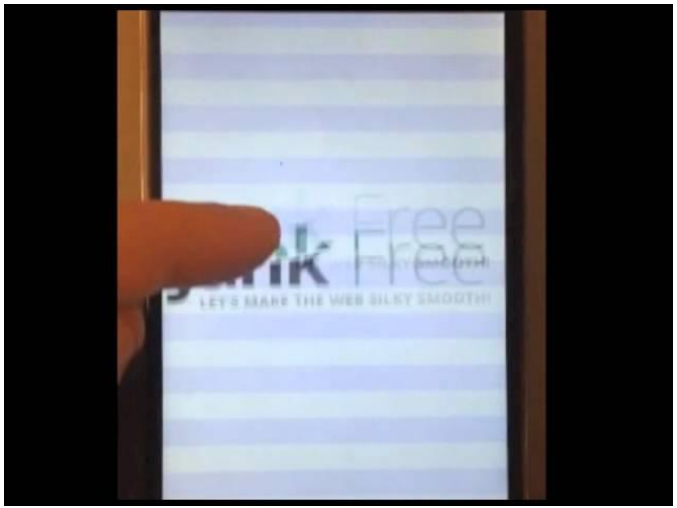
What makes rendering hard:

- pixel accuracy
- spec compliance
- sandboxing

Web Rendering Through the Ages

1989-2007:

- Software rendered
- 18 years of accumulated rendering features
- Highly influenced by CPU raster capabilities
- Most of focus on page load time, not rendering
- Documents that scroll

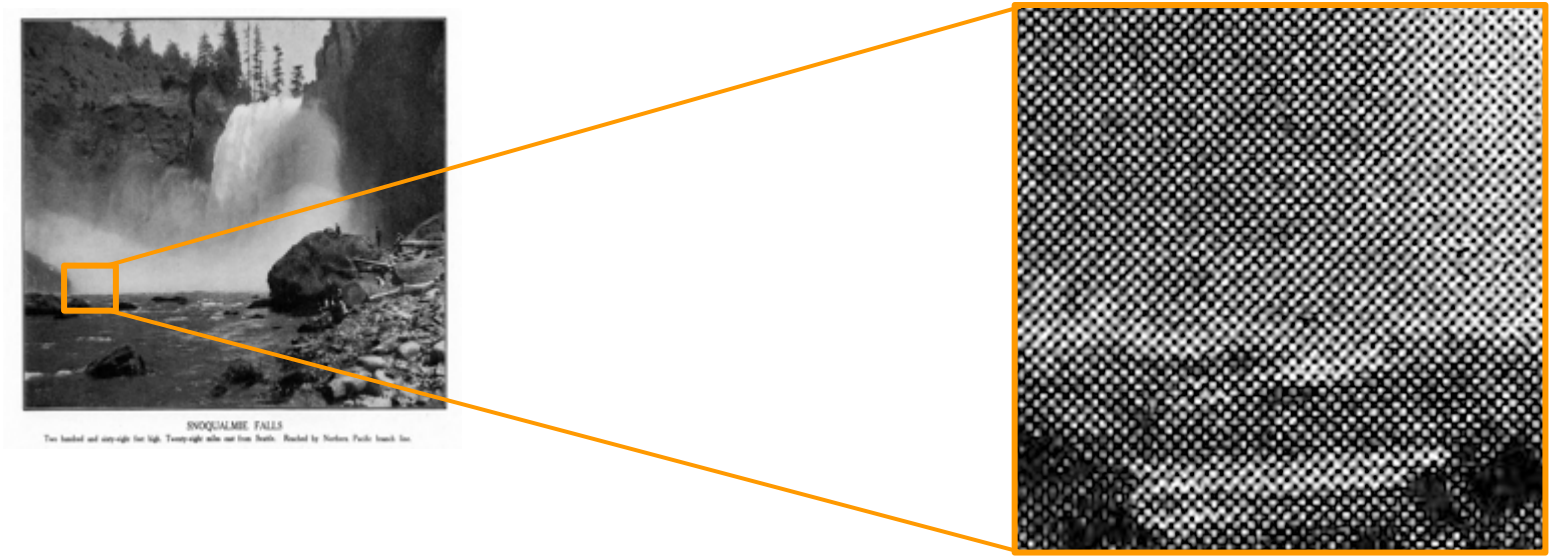


A Web Gem….

```
{
  width: 0px;
  height: 0px;
  border-style: solid;
  border-width: 0 10px 10px 10px;
  border-color: transparent transparent
                #007bff transparent;
}
```

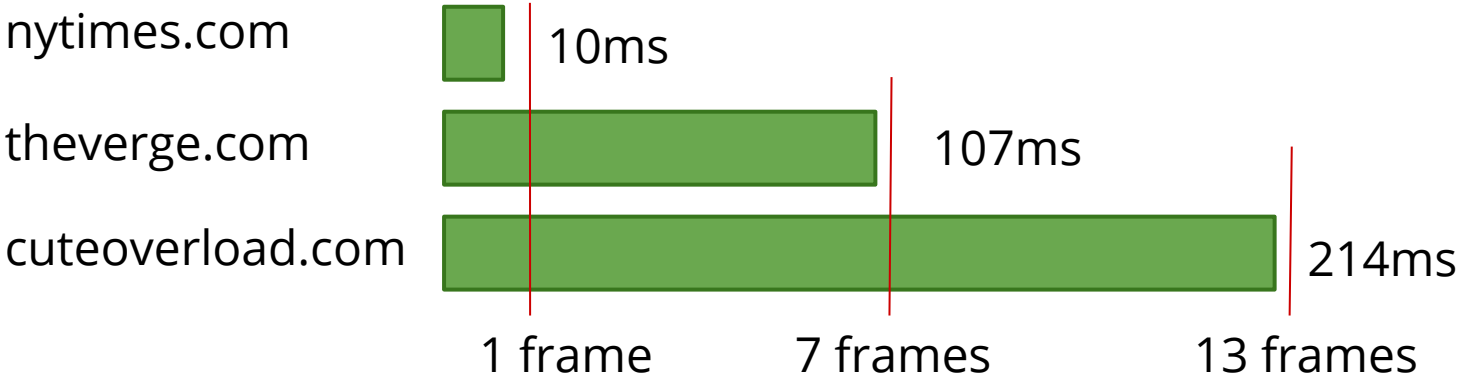
Another Web Gem

```
<!-- native size of foo.jpg is 4594x3770 -->
<img src="foo.jpg"
     width="100px" height="100px">
```



So as you would expect….

From-scratch software rasterize time is agonizingly slow:



(Nexus 10)

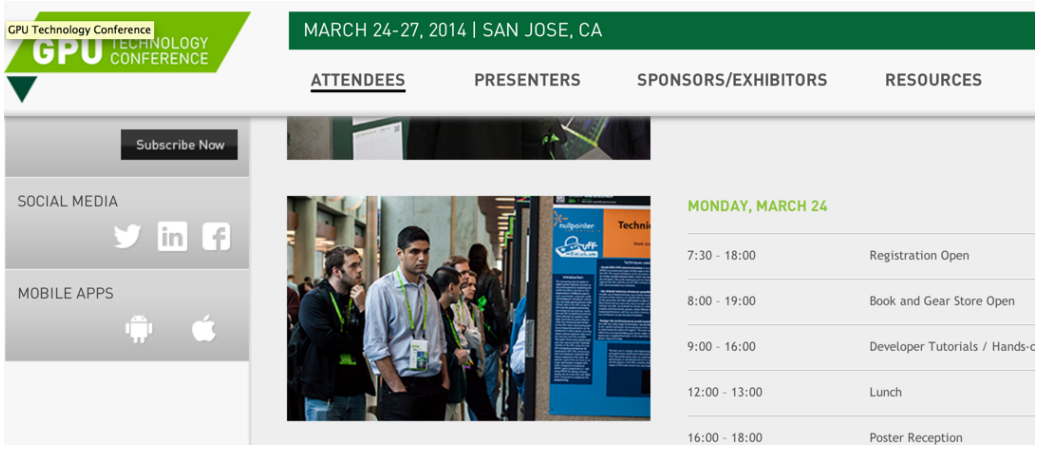
Web Rendering Through the Ages

2007-present:

- iOS Safari led the charge

- Assume the web page is incurably janky….

Layers + Compositing FTW

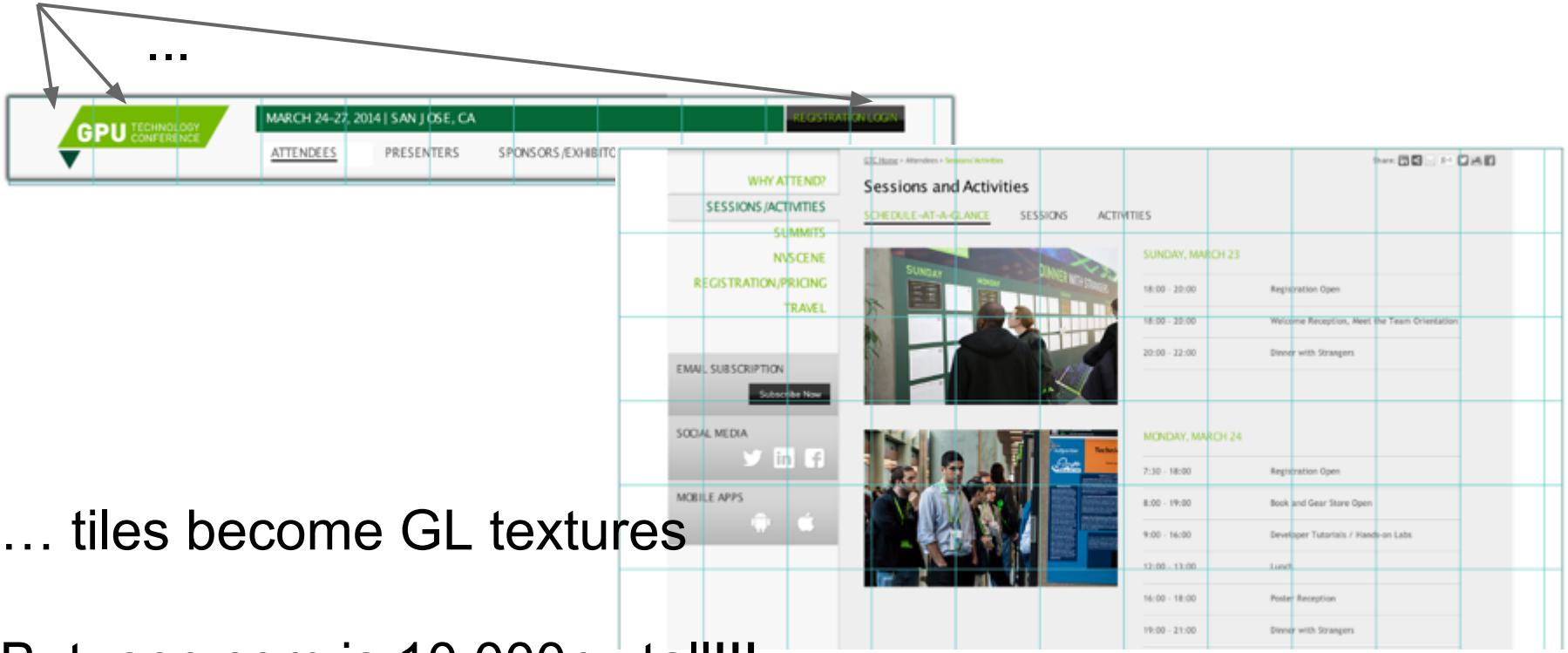


Layers created for….

- fixed position
- scrolling elements
- animations
- will-change hint (*)

… lots more

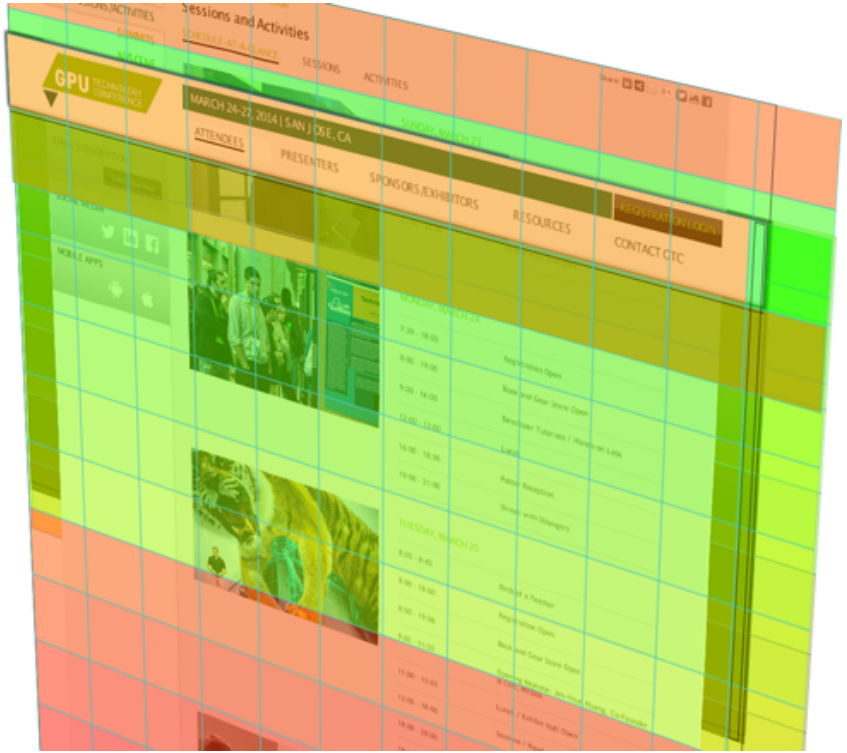
Layers are diced up into tiles..



… tiles become GL textures

But, cnn.com is 10,000px tall!!!

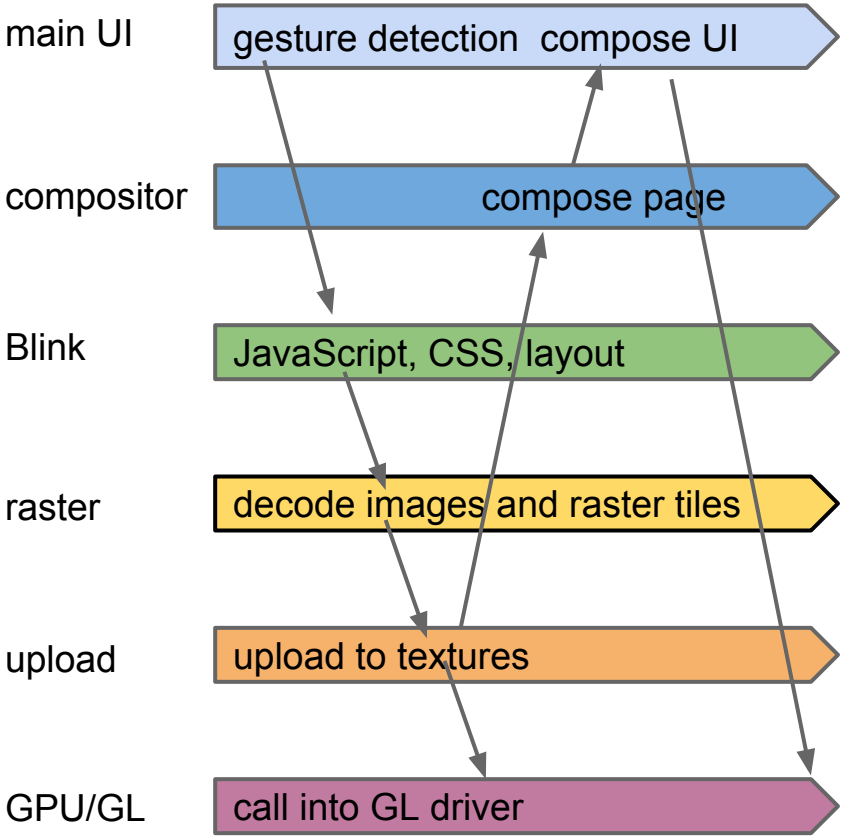
… then prioritized and rasterized



so…..

layers + tiles + compositing is enough, right?

The Naive Way to Scroll



- The good: This is how the web defines things as happening.
- The bad part: so many places things can go wrong…

Faster: Threaded scrolling

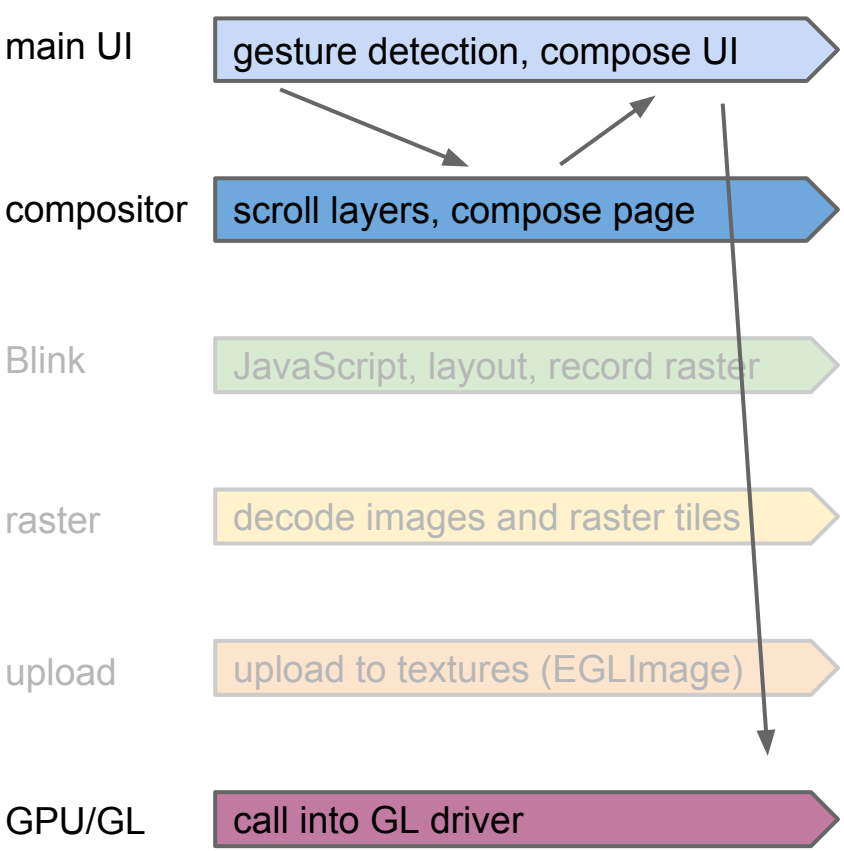
optimize for the most important cases

- scrolling
- pinch

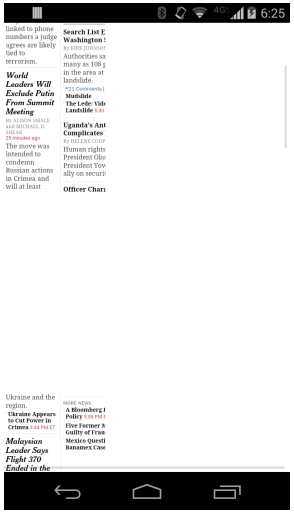
introduce dedicated composting thread

checkerboard when the user exposes things that are not ready

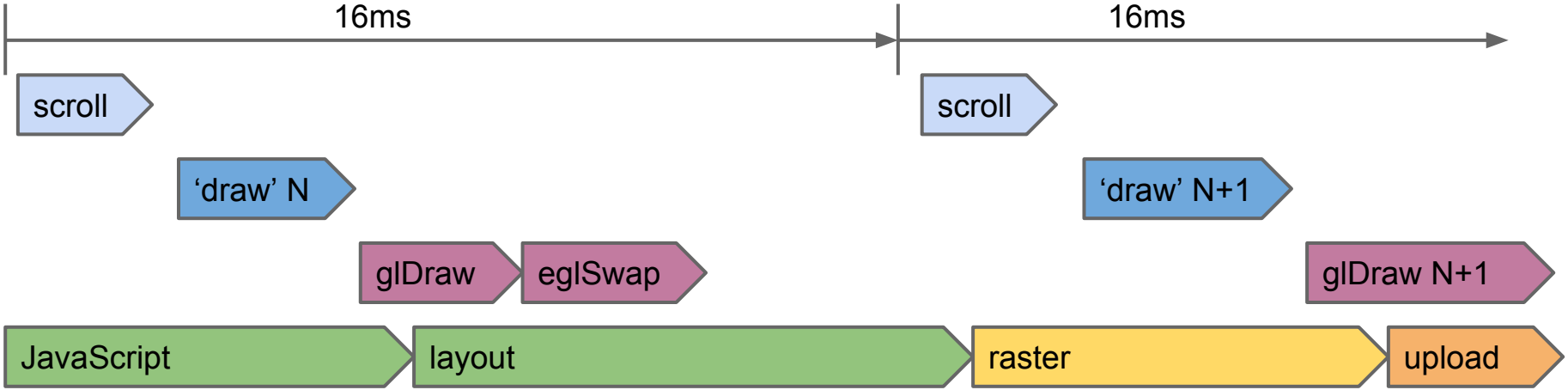
Fast Path Scrolling



Scroll with potential checkerboarding while Blink thread works on future frames.



Page is fully responsive to touch with potentially some missing tiles. In parallel, layout and rasterization takes place for future frames.



Compositor Thread Animations

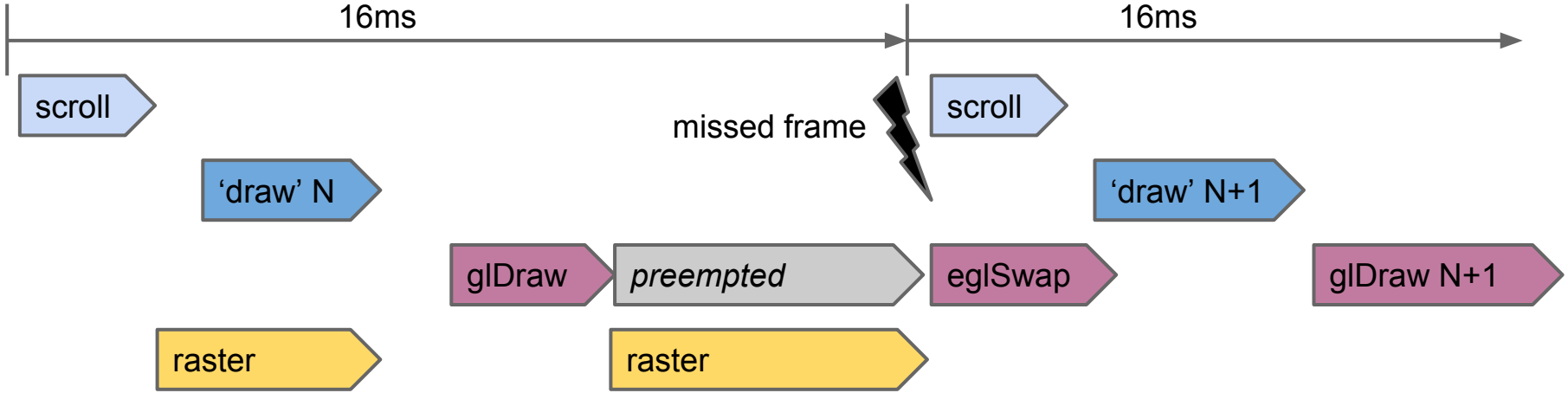
Compositor thread can do more than scrolling:

- opacity
- transforms
- pixel filters

http://jankfree.org/examples/smooth_simple_with_trace_markup.html

Multithreading realities

Great on powerful desktop, but more fragile on mobile due to fewer cores and power management. Threads can get descheduled with the pipeline latency increasing.



Beyond

From 2007 until recently, we've been focused on compensating for jank

This makes the web kinda broken: we're bypassing the web with fast scrolling, pinch

Conceptual good things

- So this leads us to thinking about GPU rasterization [and many other things]

- Sounds good in concept...
  - Better throughput when full screen changes
  - Avoids desched problems

but….

Another Web Gem!

```
{
    width: 50px;
    height: 50px;
    background-color: #fff;
    border-style: solid;
    border-color: #ff0000 #00ff00
                  #0000ff #ff00ff;
    border-width: 16px 24px 32px;
    box-shadow: 4px;
    border-radius: 10px;
    padding: 12px 14px;
    box-shadow: 10px 10px 4px #c2c2c2,
        inset 0px 0px 10px 10px #00c0c0;
}
```

More web gems

hairline with dashing

Our approach: hybrid rasterization

- Rasterize with GPU, but only when it makes sense
- Lots of heuristics (like when we make layers!)
  - will-change & animation: strong signals to use GPU raster
  - Vetoing: this content has something whacky, fallback to CPU

Pinch zoom...

We consider pinch zoom solved:

- Crisp-up when there are CPU resources
- Blurry when needed to preserve frame rate

Pinch zoom is a content bug: the page should have been mobile friendly in the first place.

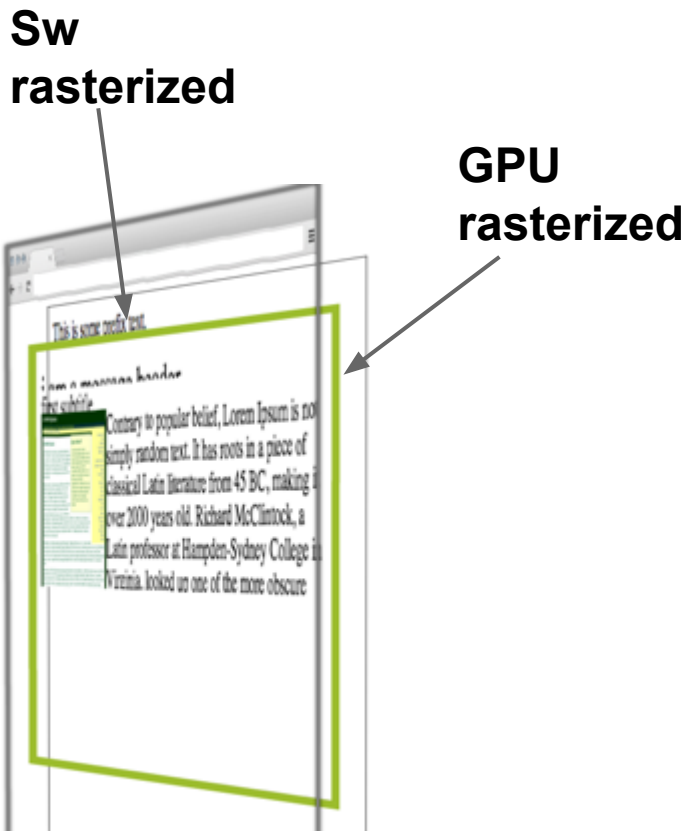
An example

This is some prefix text.

i am a message header

first subtitle

Contrary to popular belief, Lorem Ipsum is not simply random text. It has roots in a piece of classical Latin literature from 45 BC, making it over 2000 years old. Richard McClintock, a Latin professor at Hampden-Sydney College in Virginia, looked up one of the more obscure Latin words, consectetur, from a Lorem Ipsum passage, and going through the cites of the word in classical



http://jsfiddle.net/jx5De/6/

Going fast

98% of what we get are simple rects & drawText

- Most paths end up being rounded-rect corner cases

Biggest concerns with GPU raster are:

- Overdraw
- Clipping
- Image uploads
- GLES2 limitations

Onward!

- Re-explain rendering in GPU friendly terms

- Keep minimizing power draw
- Make content viable without the threads

- Integrate more game engine and rendering tech

Questions?

Nat, nduca@chromium.org,

Daniel, sievers@chromium.org