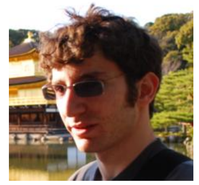
Parallel Webpage Layout

Berkeley Parlab

Leo Meyerovich, Ras Bodik



Leo Meyerovich

Ras Bodi

Browsing the Web on 1 Watt: Algorithms for Parallelizing Cascading Style Sheets

Going Smaller Isn't Easy

SunSpider **micro** benchmarks

iPhone

44 seconds

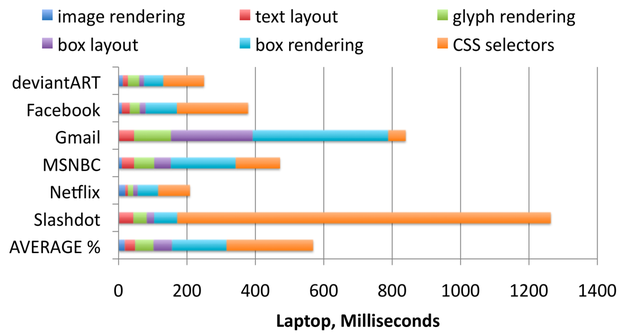
MacBook Pro

3 seconds

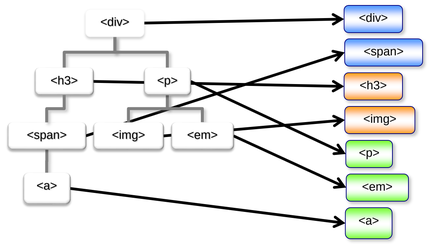
iPhone:
http://arstechnica.com/apple/news/2009/03/javascript-to-get-3x-speed-boost-in-iphone-os-30.ars

rule of thumb: 10x slowdown

Browsers are CPU Bound

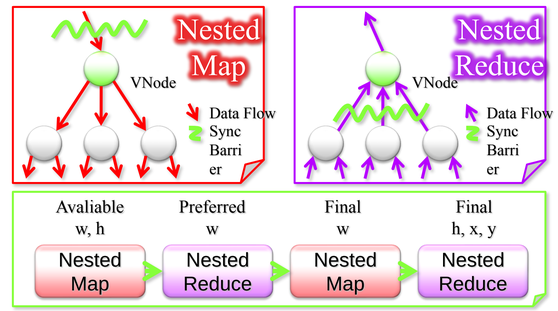


2. Cascading



(parallel for)

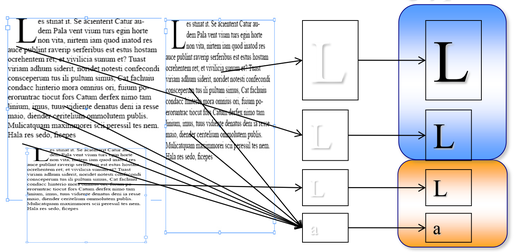
5. Parallel Layout



(parallel speculative attribute grammars)

4. Font Handling

Determine metrics and render glyphs

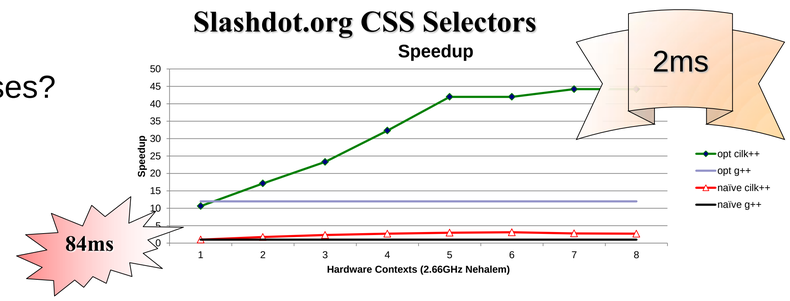


(parallel reduction then parallel for)

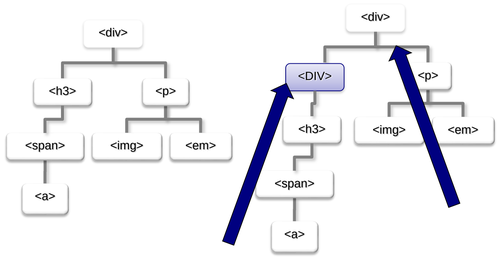
Parallelizing Layout

- Browser-like systems are crucial but slow
  - Go from 1.5s to 100ms on laptops?
  - Go from 10s to 1s on handhelds?
  - Tablet? E-book? Watch? Contact lenses?
- CPU is increasingly the bottleneck
  - >50% of CPU time is in layout
  - Parallelism!
- Basic Algorithms for:
  - Templating 1: **rule matching**
  - Templating 2: **cascading**
  - Normalization (+ translation to BSS language)
  - Font metrics and redering
  - Layout: boxes and text
  - Rendering: painting and compositing

1. CSS Selector Matching



3. Normalization/Translation



(bottom-up tree traversal)

6. Parallel Rendering

Same as 5 (parallel layout)

Next

more implementation
GPU rendering
incrementalization
automation