Mozilla & Firefox Quantum

Mozilla | 12 Mars 2018

Bonjour !

Je suis Sylvestre Ledru

Je parle de Firefox

Twitter @SylvestreLedru

Who Am I?

37 yo

Curriculum

➔ DEUG MIAS (first year) - Orléans
➔ DUT informatique de gestion
➔ IUP MIAGE
➔ DESS informatique distribuée (Paris XII)

Who Am I? Before Mozilla

➔ PhD in Australia & Paris XII (3 months)

➔ Worked for two years in a Geophysics company in Melbourne

Who Am I? Before Mozilla

- ➔ Inria on Scilab (+ Digiteo + Scilab Enterprises) for 7 years
- ➔ Irill for 4 years
- ➔ Debian for 11 years – LLVM/Clang for 5/6 years

Who Am I? Before Mozilla

- ➔ Mozilla for 4 years – Release manager
- ➔ Lead of the release management & stability teams – 15 people
- ➔ Head of the Mozilla French branch

About:

moz://a

About:Mozilla

- ➔ Adventure started by Netscape (~1994)
- ➔ Failed against Microsoft (Internet Explorer)
- ➔ Decided to open the sources of Netscape (20 years ago)

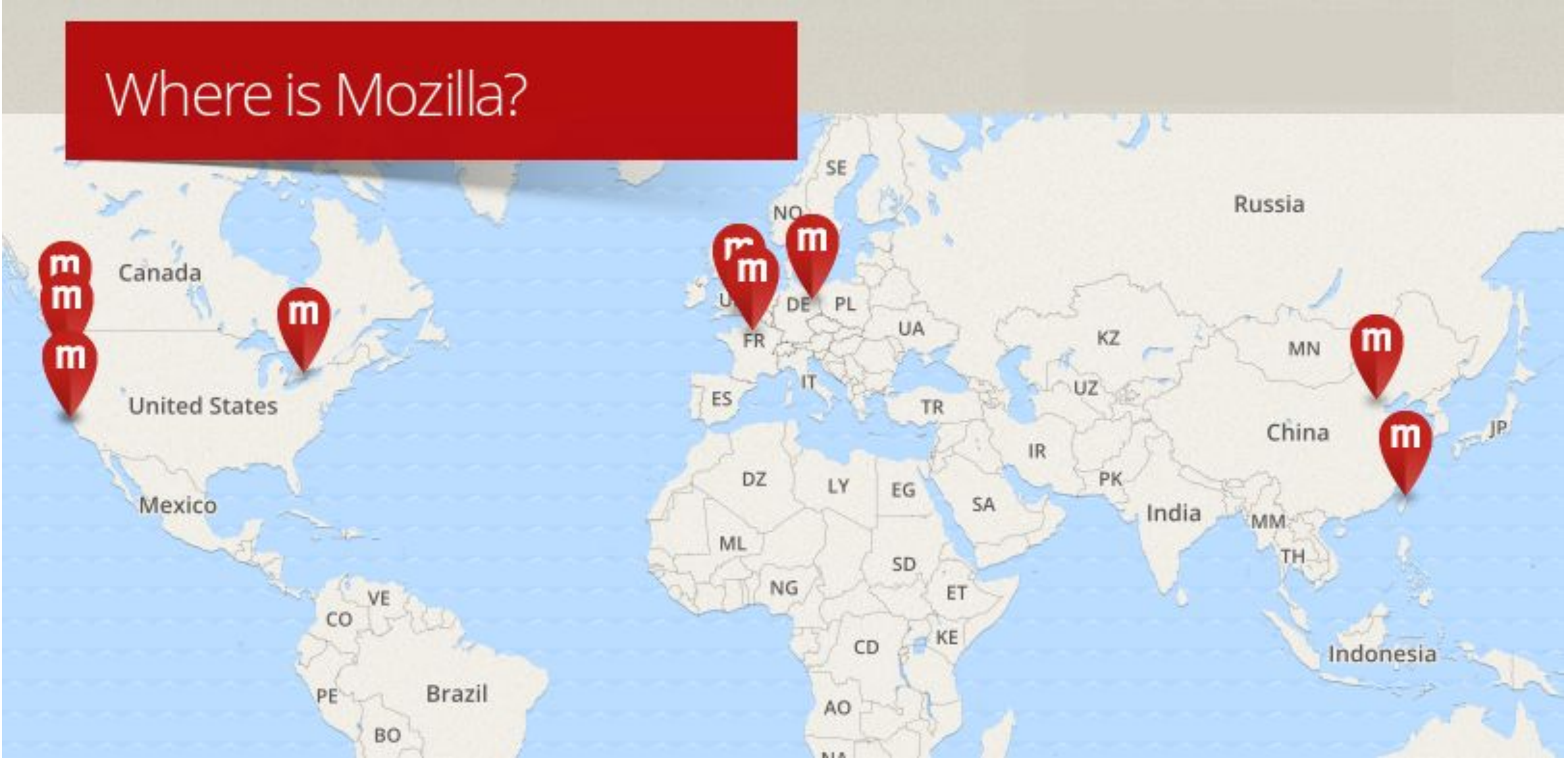
- ➔ Documentary about this period:
  Code rush:
  https://www.youtube.com/watch?v=u404SLJj7ig

About:Mozilla

- ➔ Brought by AOL in 1998
- ➔ AOL gave some money to the Mozilla Foundation in 2003
- ➔ Mozilla was nothing
- ➔ Massive refactorings
- ➔ Firefox 1.0 released 12 years ago
  A game changer (popup blocker, tab, etc)

About:Mozilla

- ➔ About 1100 employees - 11 offices



About:Mozilla

- Revenue of US$423M (2015)
  - Mainly from the search deal
  - Google before 2015, then Yahoo and others and now Google and others

- Salaries ?
  - Important competition with Fb, Twitter, Google, etc
  - Bonus

About: Firefox

moz://a

About:Firefox

- ➔ Web browser with ~500 million users
  - ◆ Only (major) browser developed by a non-profit

- ➔ Support 4 operating systems:
  - ◆ Microsoft Windows XP => 10 (32 & 64 bit)
  - ◆ GNU/Linux (32/64)
  - ◆ Mac OS X (64)
  - ◆ Android (various arm + intel)

- ➔ iOS – not based on Gecko

About:Firefox

- Second or third browser in term of market share
  - 12 to 22 % market share
  - (don't trust them too much)

- Chrome started in 2008 by former Mozilla contributors
  - Current leader

About:Firefox

- We release every 6 to 8 weeks

- 7 major releases published (one ESR) in 2017
  - Including Firefox Quantum 57
  - + 26 minor releases

About:Firefox:Releases

- ➔ Other versions:
  - ◆ In parallel, 2 other feedback branches :
  - ◆ Nightly - updated daily with recent code changes
  - ◆ Beta – 2 per week Desktop – 1 for Mobile
  - ◆ Devedition
- ➔ Managed by 6 people

About:Firefox:ESR

- Major ESR (Extended Support Release) version is created every ~12/13 months, with two cycles of overlap with the prior version
  - Security (high/critical) every 6 weeks (in sync with Firefox)
  - Other issues by request from mailing list community
  - Used by some GNU/Linux distributions (example: Debian), major companies, universities, etc
  - Active community mailing list, various sized orgs
  - Base for Thunderbird & Seamonkey
- Firefox 60 will introduce group policy controls

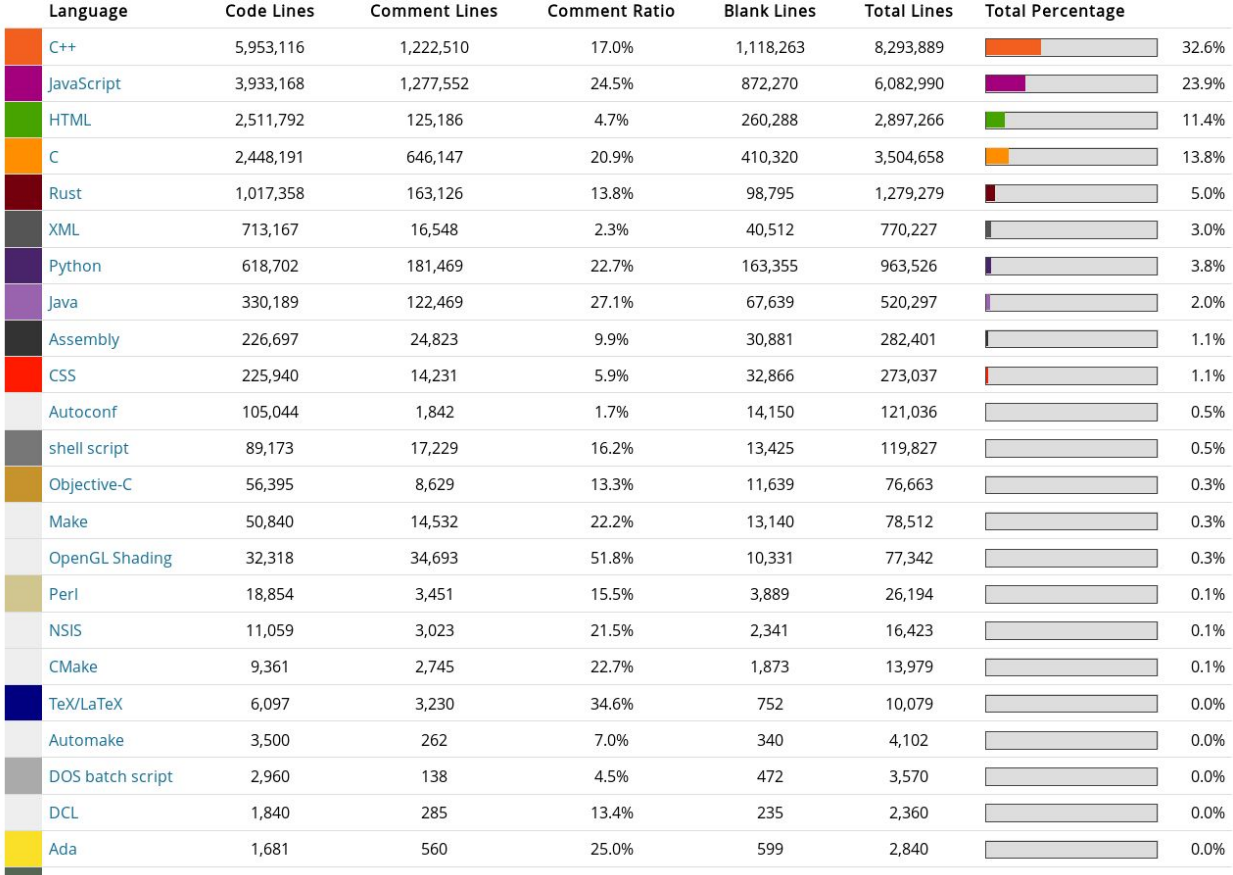
About:Firefox:code

- ➔ Gecko is the based of Firefox
  - ◆ And ... Thunderbird, Seamonkey and Firefox OS (rip)

- ➔ ... has had 407 227 commits made by 5 482 contributors representing 18 374 566 lines of code

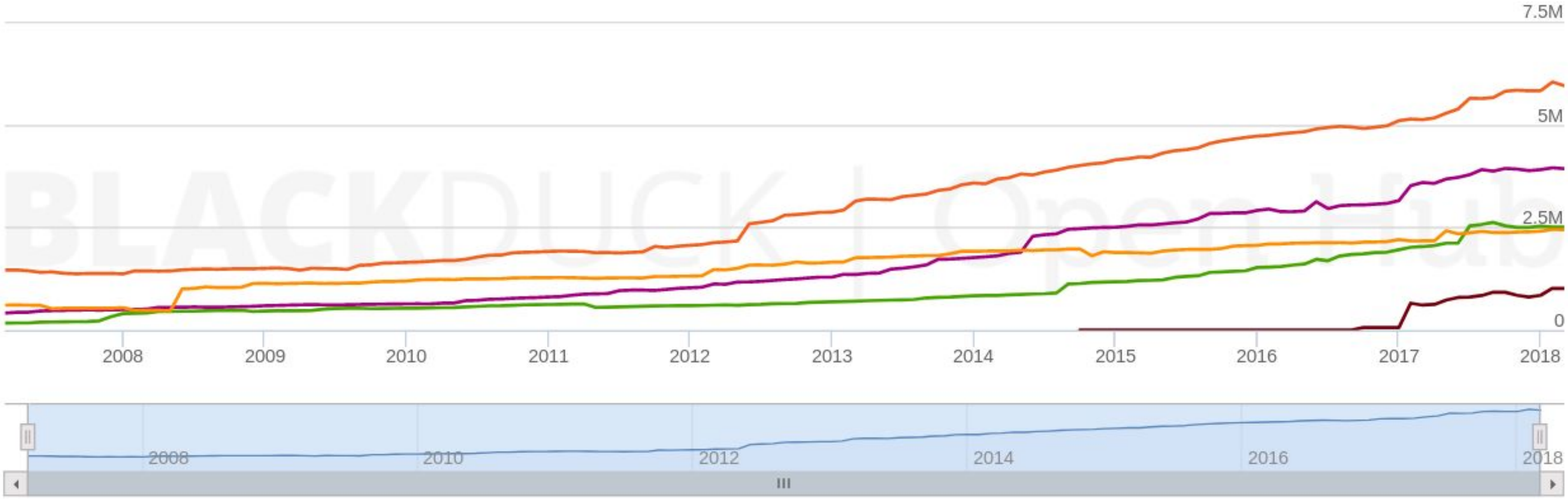
- ➔ Average of 440 developers / month
- ➔ 1262 contributors over last year

About:Firefox:code

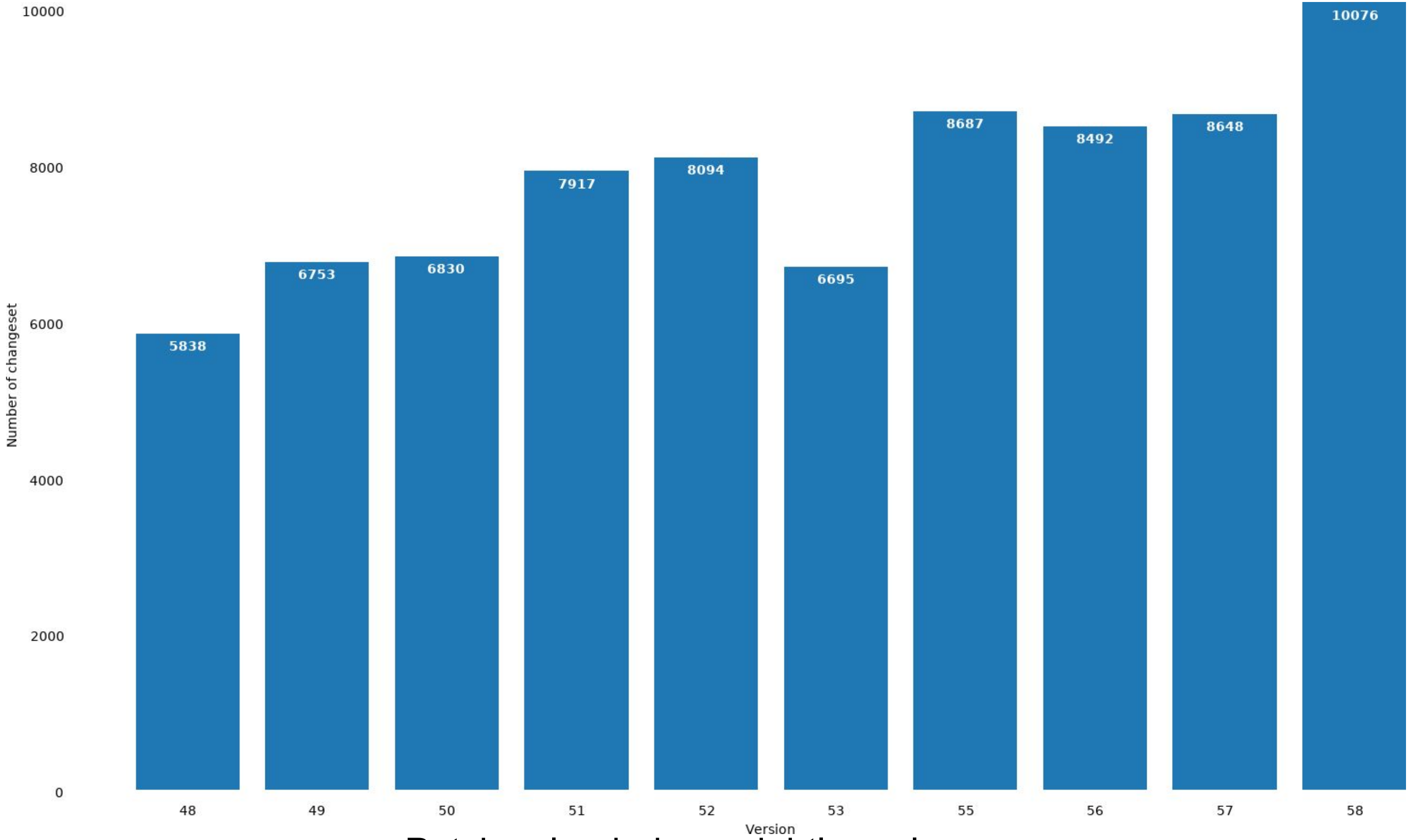


Language	Code Lines	Comment Lines	Comment Ratio	Blank Lines	Total Lines	Total Percentage
C++	5,953,116	1,222,510	17.0%	1,118,263	8,293,889	32.6%
JavaScript	3,933,168	1,277,552	24.5%	872,270	6,082,990	23.9%
HTML	2,511,792	125,186	4.7%	260,288	2,897,266	11.4%
C	2,448,191	646,147	20.9%	410,320	3,504,658	13.8%
Rust	1,017,358	163,126	13.8%	98,795	1,279,279	5.0%
XML	713,167	16,548	2.3%	40,512	770,227	3.0%
Python	618,702	181,469	22.7%	163,355	963,526	3.8%
Java	330,189	122,469	27.1%	67,639	520,297	2.0%
Assembly	226,697	24,823	9.9%	30,881	282,401	1.1%
CSS	225,940	14,231	5.9%	32,866	273,037	1.1%
Autoconf	105,044	1,842	1.7%	14,150	121,036	0.5%
shell script	89,173	17,229	16.2%	13,425	119,827	0.5%
Objective-C	56,395	8,629	13.3%	11,639	76,663	0.3%
Make	50,840	14,532	22.2%	13,140	78,512	0.3%
OpenGL Shading	32,318	34,693	51.8%	10,331	77,342	0.3%
Perl	18,854	3,451	15.5%	3,889	26,194	0.1%
NSIS	11,059	3,023	21.5%	2,341	16,423	0.1%
CMake	9,361	2,745	22.7%	1,873	13,979	0.1%
TeX/LaTeX	6,097	3,230	34.6%	752	10,079	0.0%
Automake	3,500	262	7.0%	340	4,102	0.0%
DOS batch script	2,960	138	4.5%	472	3,570	0.0%
DCL	1,840	285	13.4%	235	2,360	0.0%
Ada	1,681	560	25.0%	599	2,840	0.0%

About:Firefox:code



About:Firefox:Code



Patches landed per nightly cycle

About:Firefox:Continuous Integration

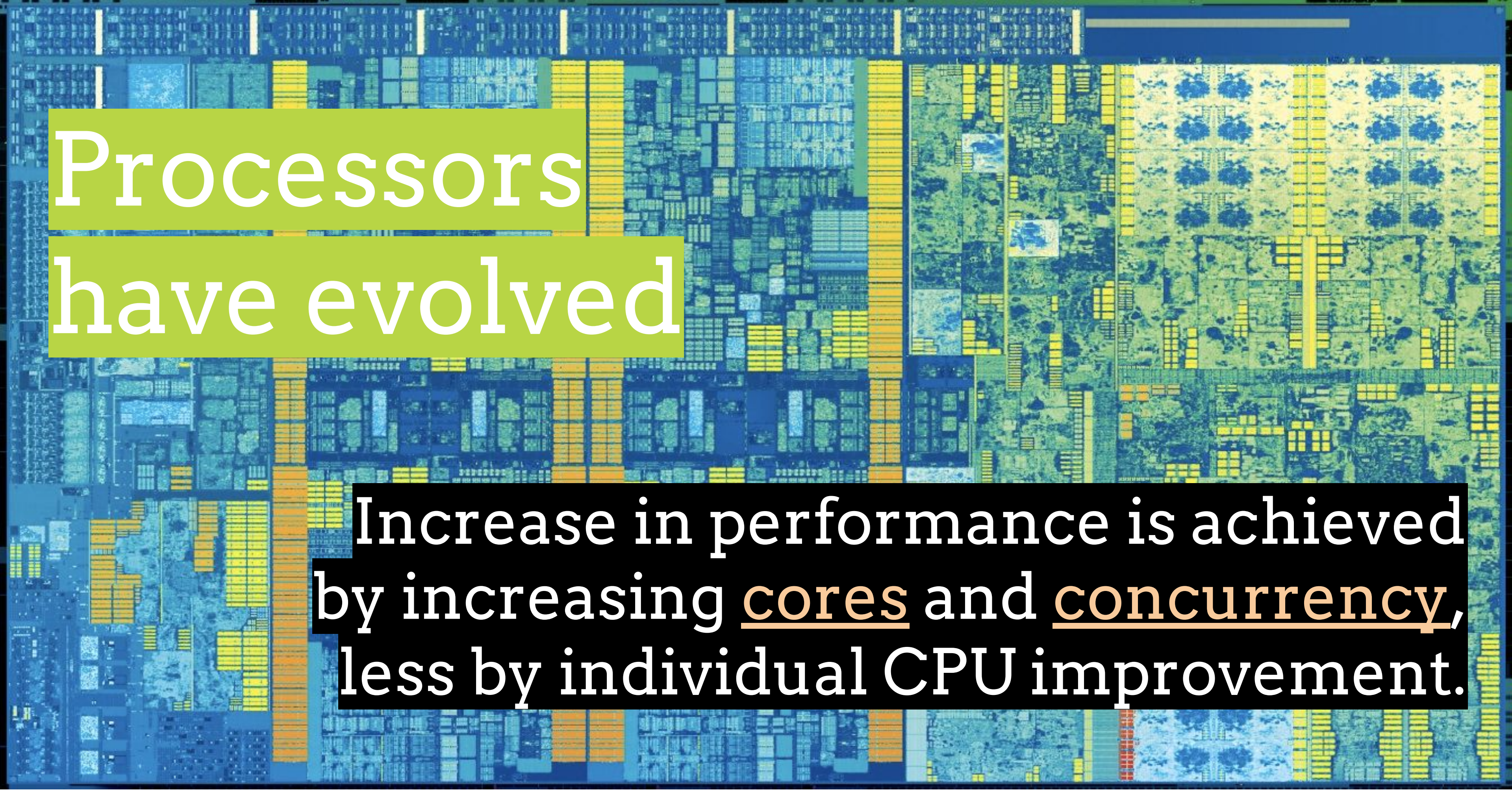
- We run a few tests... with a few different platforms and options
- 1 506 hours for the average full CI run
- Numbers from November 2017:
  - 8 319 189 tasks
  - 299.8 machine years
  - 927 333 unique machines

About:Firefox & release mgmt

- Making sure that Firefox releases are released in good shape and on time, deciding when to build & when to wait for patches.

- Coordinate work between :
  - Firefox Developers
    Quality Engineering
  - Stability
  - Release Engineering
  - EPM
  - Product
  - L10n
  - Security
  - User Advocacy
  - Addons
  - Marketing
  - …

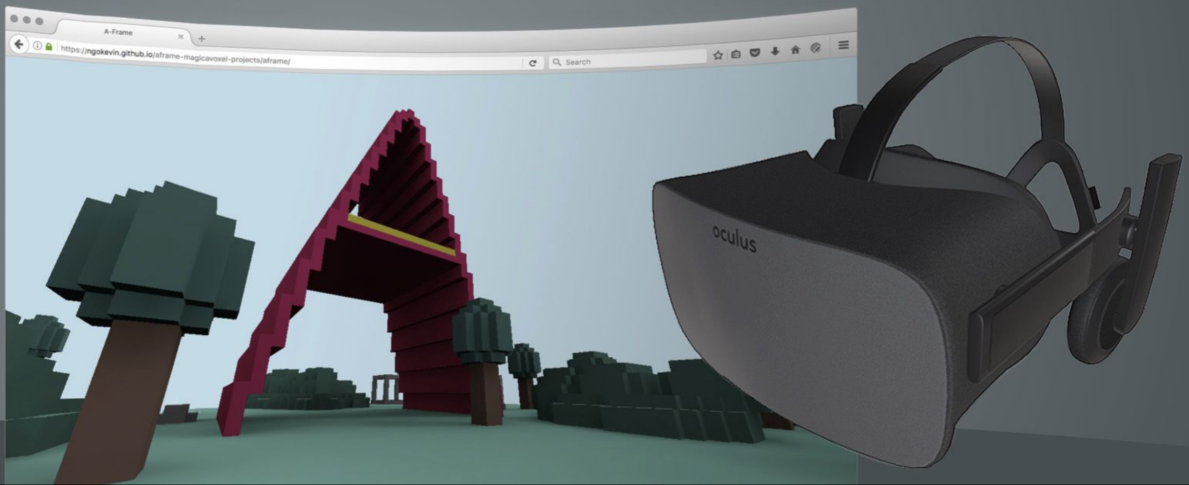
Thanks to Jean-Yves Perrier for the Quantum's slides



Processors have evolved

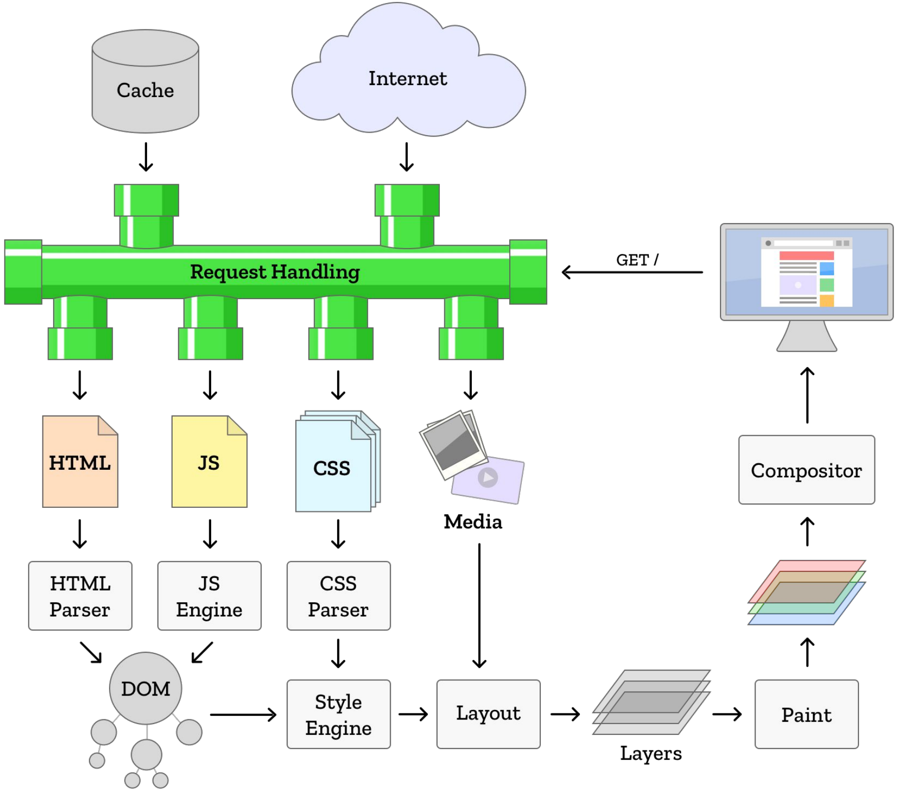
Increase in performance is achieved by increasing cores and concurrency, less by individual CPU improvement.

Expectations changed


A-Frame
https://ngokevin.github.io/aframe-magicavoxel-projects/aframe/
Search
oculus

In the 2000s we wanted pages with images; today we want 60 fps virtual reality with HD videos.

The rendering engine



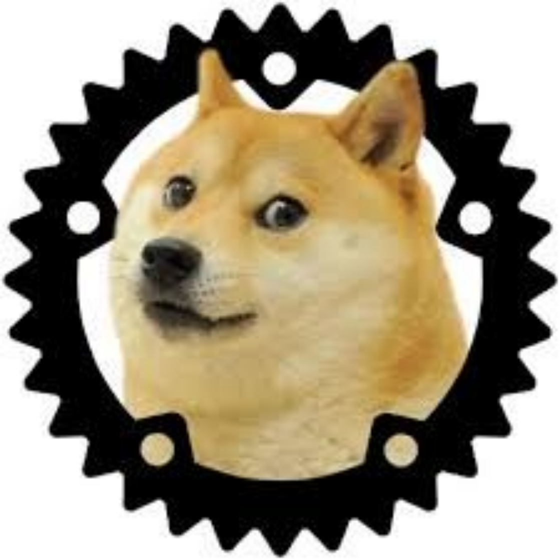
Drawing by Potch

Our testbed: Servo

An experimental engine:

- Written in Rust
- Massive parallelism
- No significant UI
- Break the Web

Hello World!



Project Quantum

Problems to solve:

- Stability
- Old theme
- UI jank
- Integration of improvements from Servo

Project Quantum

Quantum Compositor

Quantum CSS (Stylo)

Quantum Flow / Quantum DOM

Photon

Quantum Renderer (WebRender)

Quantum Compositor

GPU does composition well: offload to GPU

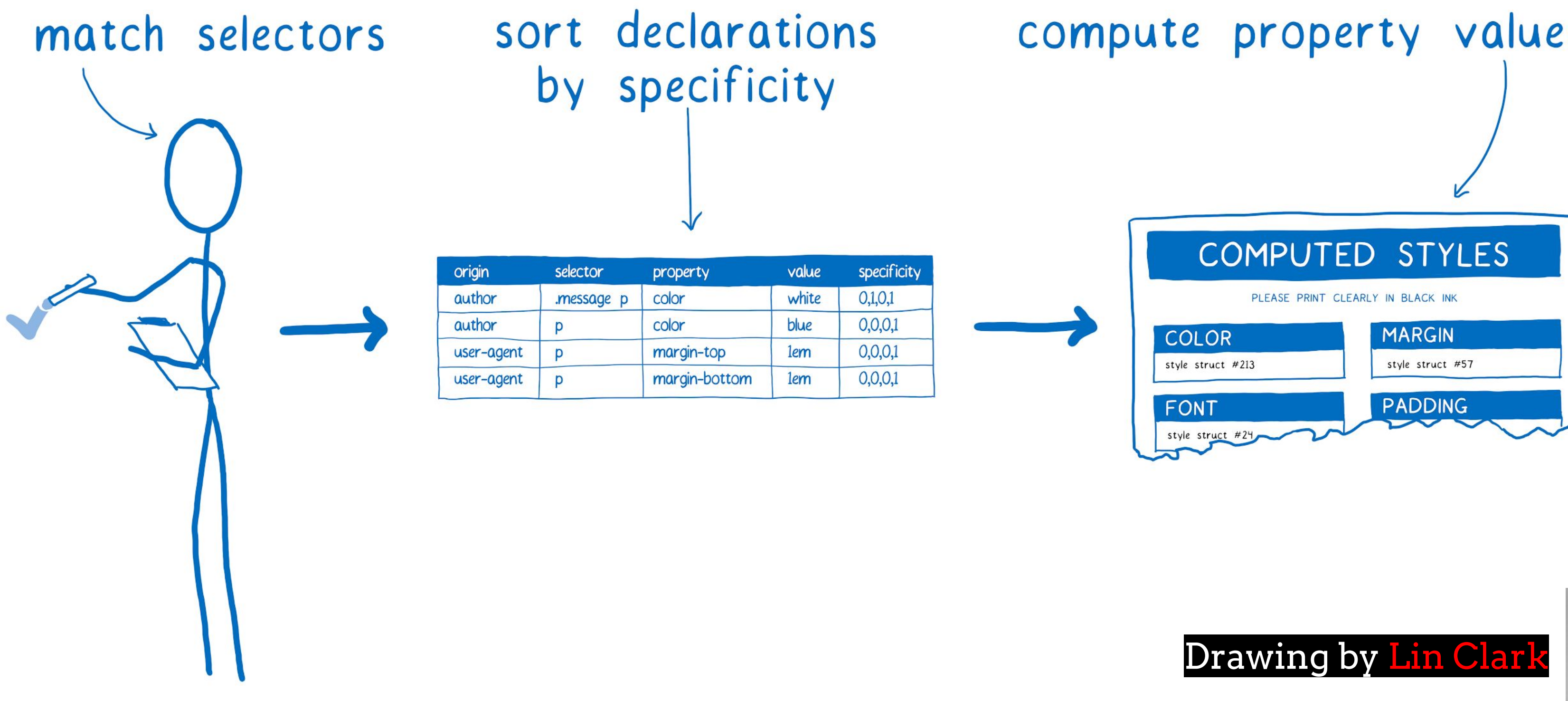
On Windows, lots of crashes caused by graphic drivers

Idea: isolate the compositor in its own process

Improve: stability

Released: Firefox 53, April 19th, 2017

Quantum CSS (Stylo)



origin	selector	property	value	specificity
author	.message p	color	white	0,1,0,1
author	p	color	blue	0,0,0,1
user-agent	p	margin-top	1em	0,0,0,1
user-agent	p	margin-bottom	1em	0,0,0,1

COMPUTED STYLES

PLEASE PRINT CLEARLY IN BLACK INK

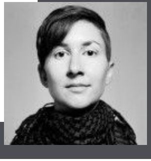
COLOR
style struct #213

MARGIN
style struct #57

FONT
style struct #24

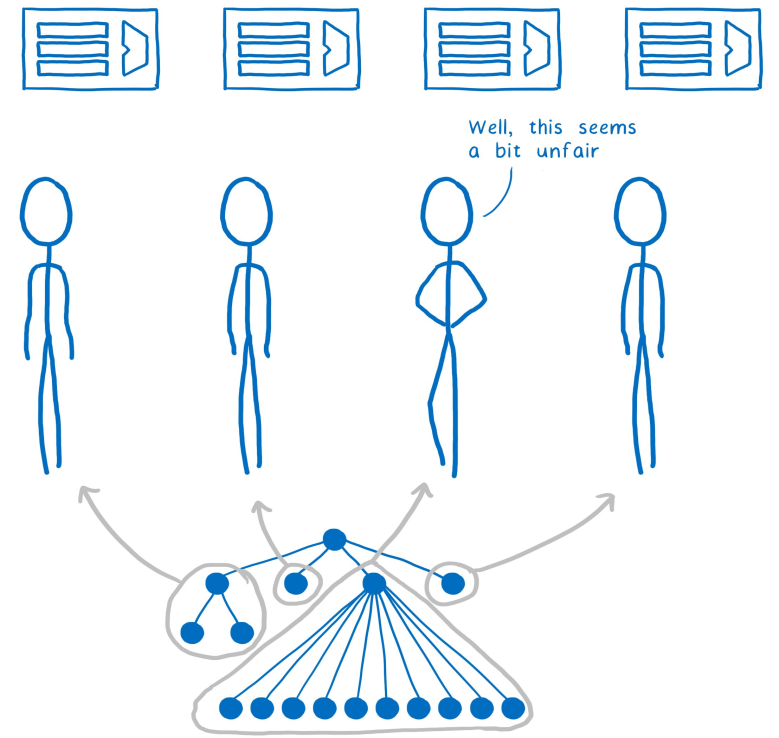
PADDING

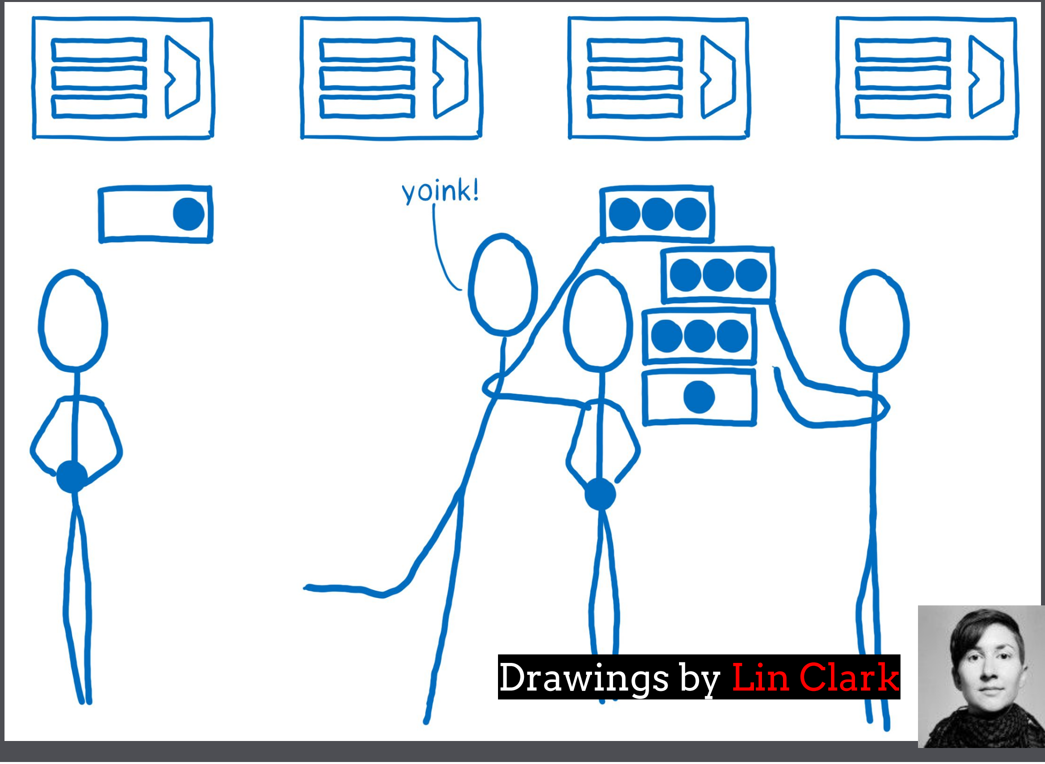
Drawing by Lin Clark



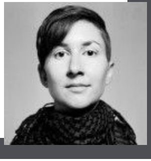
moz://a

Quantum CSS (Stylo)



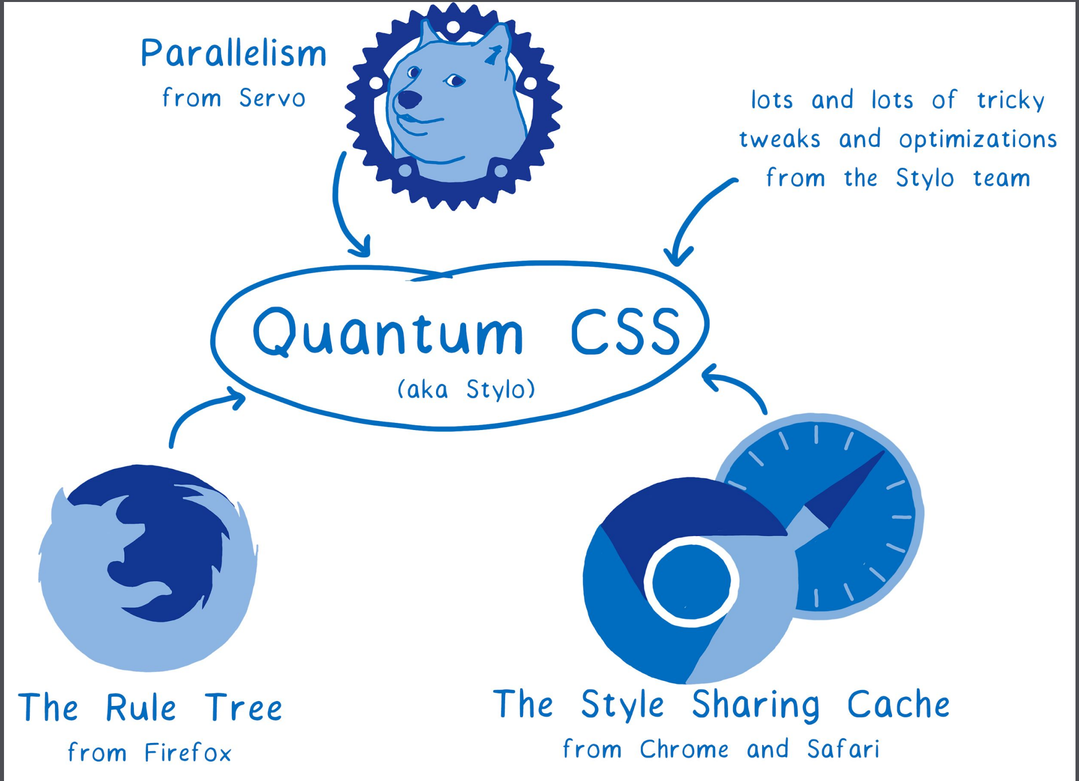


Drawings by Lin Clark



moz://a

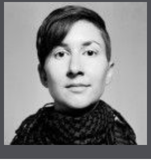
Quantum CSS (Stylo)



Style Sharing Cache

Same ids, classes, ... ?
Same inline styles ?
Same parents ?
Same “oddities”,
like :first-child()?

Drawing by Lin Clark



moz://a

Quantum CSS (Stylo)

First technology transfer from Servo

Embarrassingly parallel problem

Works best for complex pages.

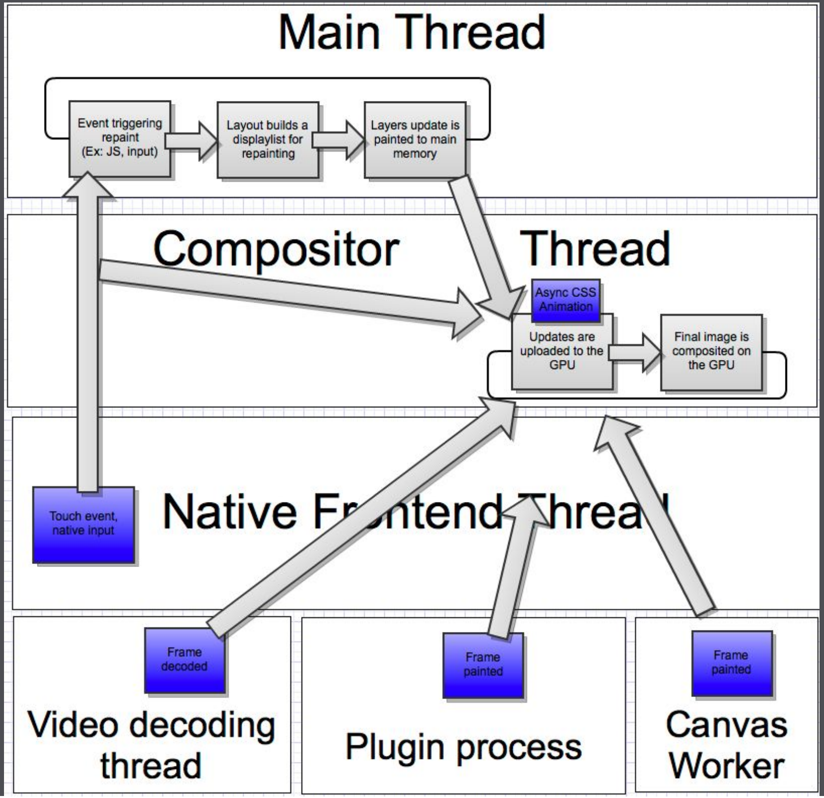
More info:
https://hacks.mozilla.org/2017/08/inside-a-super-fast-css-engine-quantum-css-aka-stylo/
by the amazing Lin Clark who did the great drawings.

Improve: performance

Desktop: in Firefox 57, November 14th, 2017

moz://a

The Main Thread Challenge


Main Thread
Event triggering repaint (Ex: JS, input)
Layout builds a displaylist for repainting
Layers update is painted to main memory
Compositor Thread
Async CSS Animation
Updates are uploaded to the GPU
Final image is composited on the GPU
Native Frontend Thread
Touch event, native input
Frame decoded
Video decoding thread
Frame painted
Plugin process
Frame painted
Canvas Worker

moz://a

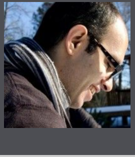
Quantum Flow

Approach performance in Firefox as a system

Measure all the things!
then triage, prioritize and (re)-measure

Goal: Fight jank

Hundreds of improvements
“Death by a million cuts”



Read Ehsan Akhgari blog posts:
https://ehsanakhgari.org/blog/2017-09-15/quantum-flow-engineering-newsletter-24

Improve: overall performance

Release: Bits by bits (Firefox 57), November 14th, 2017

QFlow: Better scrolling

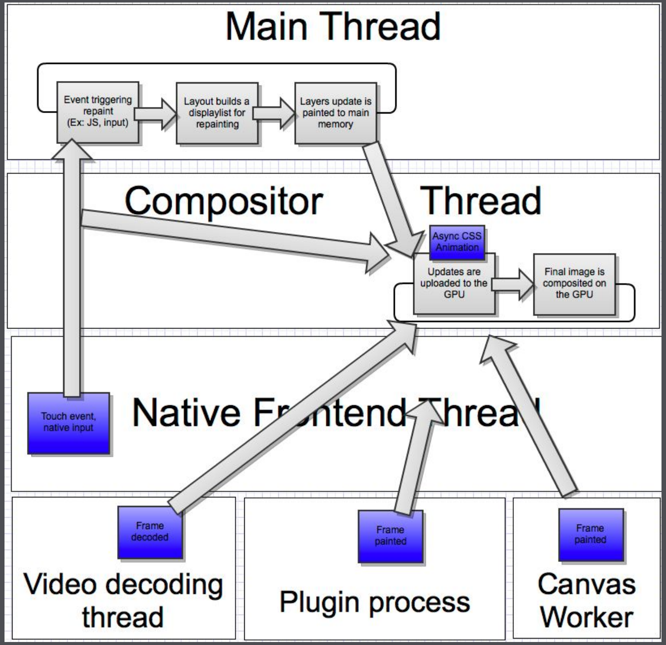
Never pause it, the user gets confused

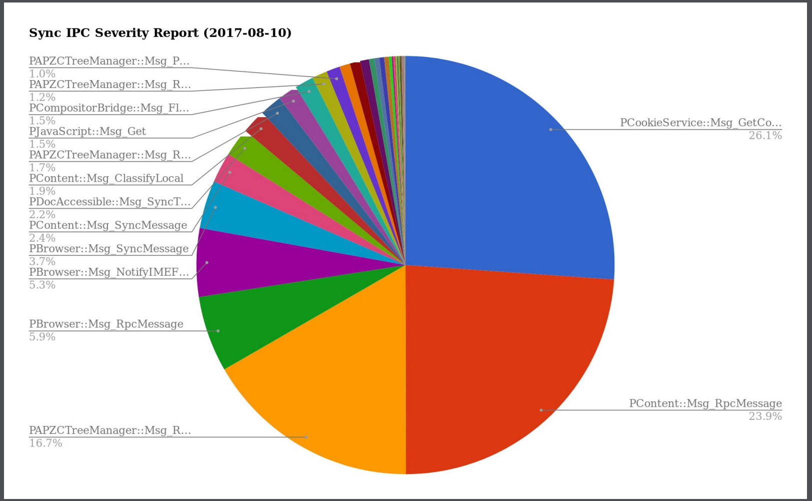
Checkerboard: When no data ready, display a default pattern.

Same overall performance, but user feeling completely different.

This was standard for mobile (touch interface),
but we now use it for all input type, even on desktop.

QFlow: No sync IPC

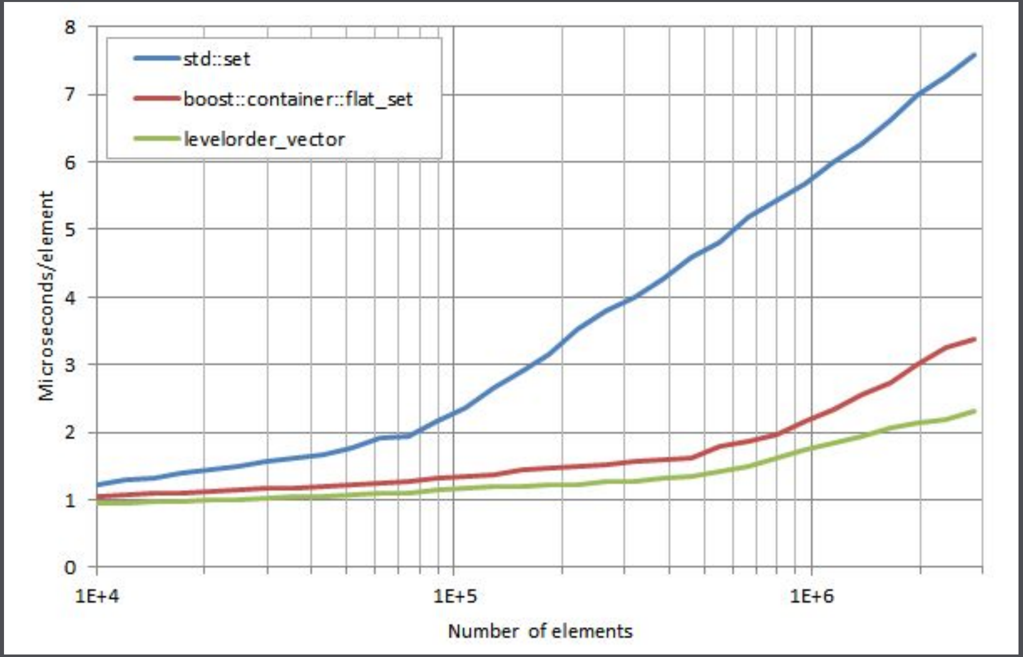

Main Thread
Event triggering repaint (Ex: JS, input)
Layout builds a displaylist for repainting
Layers update is painted to main memory
Compositor Thread
Async CSS Animation
Updates are uploaded to the GPU
Final image is composited on the GPU
Native Frontend Thread
Touch event, native input
Frame decoded
Video decoding thread
Frame painted
Plugin process
Frame painted
Canvas Worker


Sync IPC Severity Report (2017-08-10)

Annoying offenders: XUL extensions, plugins, cookies.

QFlow: Locality

E.g.
cache-friendly
binary tree search



QFlow: Controlling GC

Garbage collection is critical to memory control,
but if it is too long, it creates jank.

Evolution over the years; now it is:

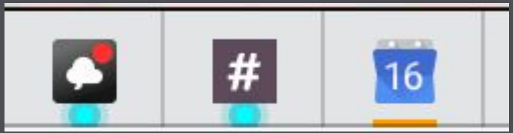
generational, to prioritize area likely to be released

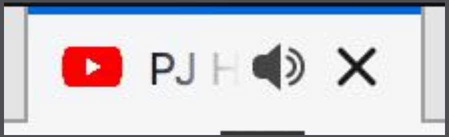
more incremental, to control how long it runs.

Background Tabs

Pause them!

Not always possible, they may do something useful:





Be defensive, and experiment.

Photon

Refresh the Firefox front end to be the user facing component of Quantum

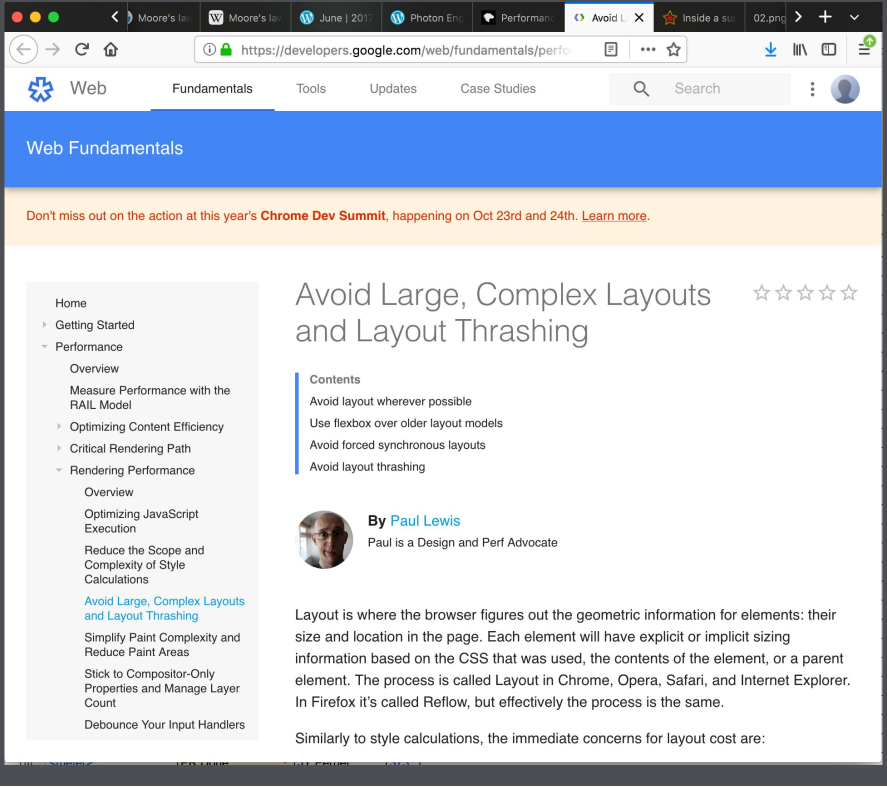
Visual Re-Design	*How it looks*
Motion	*How it feels*
Perceived Performance	*How it feels*
Structure	*How it works / Where things live*
Content	*How we talk to users*

Improve: user experience

In Firefox 57, November 14th, 2017

moz://a

Visual re-design



Don't miss out on the action at this year's **Chrome Dev Summit**, happening on Oct 23rd and 24th. Learn more.

- Home
- Getting Started
- Performance
  - Overview
  - Measure Performance with the RAIL Model
  - Optimizing Content Efficiency
  - Critical Rendering Path
  - Rendering Performance
    - Overview
    - Optimizing JavaScript Execution
    - Reduce the Scope and Complexity of Style Calculations
    - Avoid Large, Complex Layouts and Layout Thrashing
    - Simplify Paint Complexity and Reduce Paint Areas
    - Stick to Compositor-Only Properties and Manage Layer Count
    - Debounce Your Input Handlers

Avoid Large, Complex Layouts and Layout Thrashing

☆☆☆☆☆

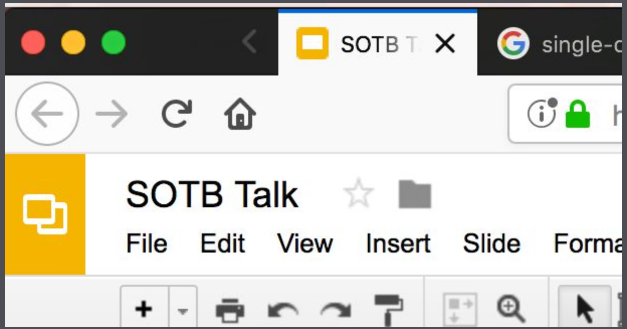
Contents

Avoid layout wherever possible
Use flexbox over older layout models
Avoid forced synchronous layouts
Avoid layout thrashing

By Paul Lewis
Paul is a Design and Perf Advocate

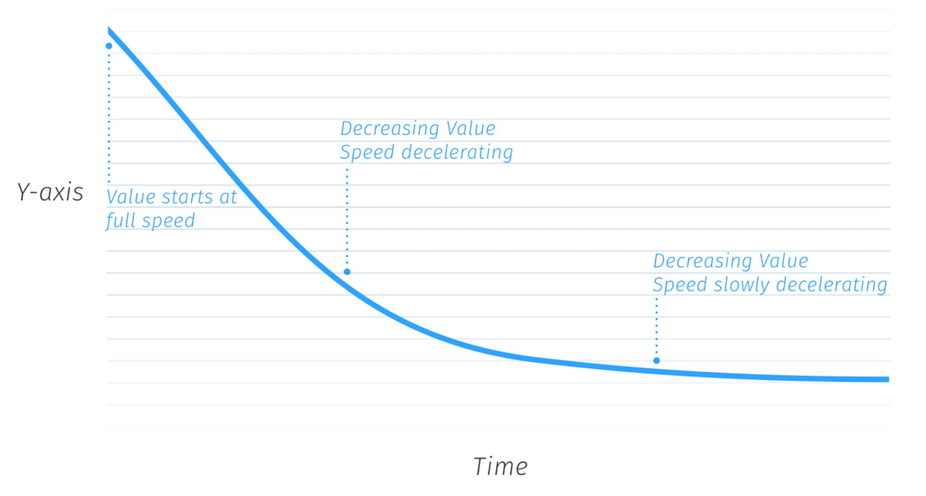
Layout is where the browser figures out the geometric information for elements: their size and location in the page. Each element will have explicit or implicit sizing information based on the CSS that was used, the contents of the element, or a parent element. The process is called Layout in Chrome, Opera, Safari, and Internet Explorer. In Firefox it's called Reflow, but effectively the process is the same.

Similarly to style calculations, the immediate concerns for layout cost are:

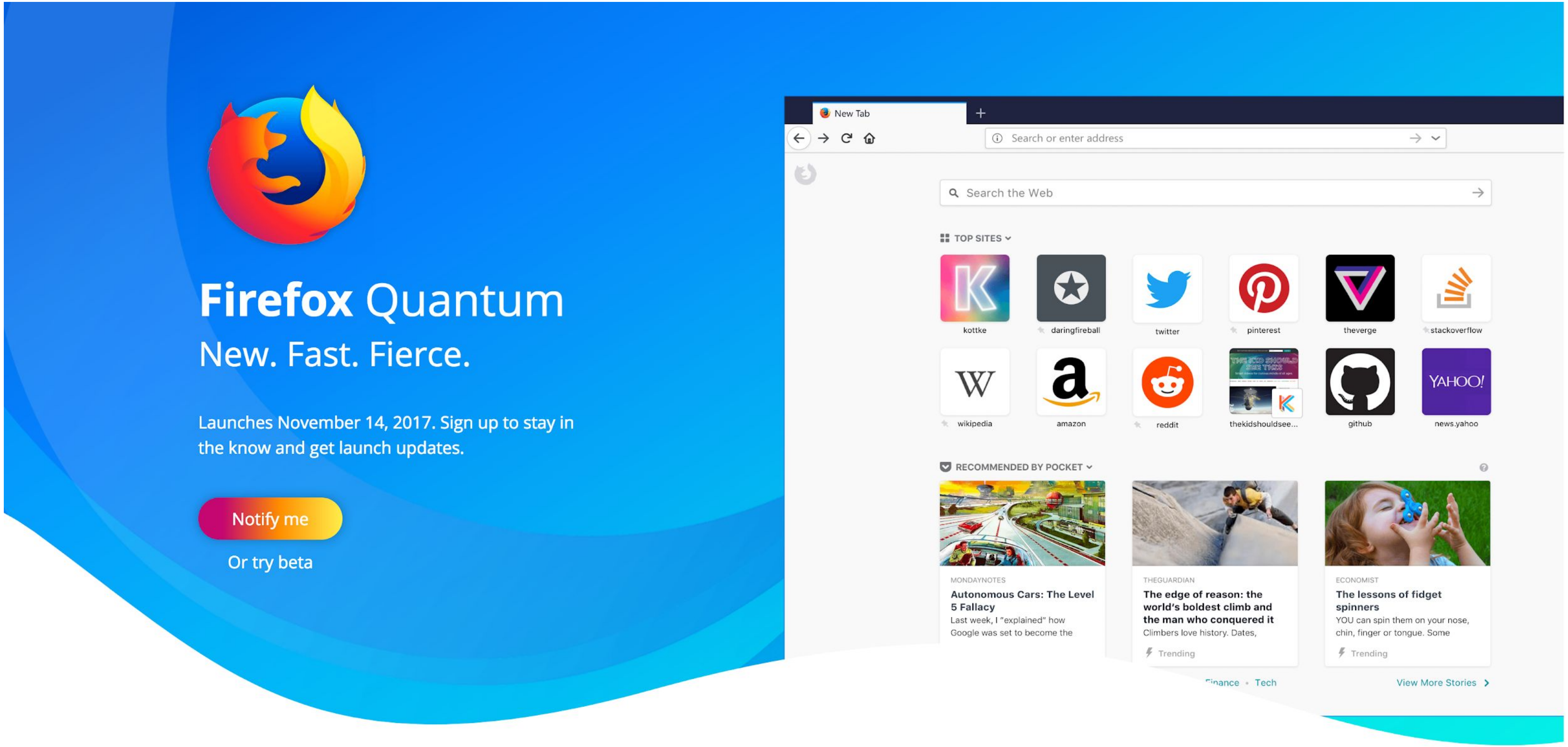


Modernizing the way Firefox looks will make it feel fast and modern.

Consistent animations



Create a consistent animation story that is both fast and responsive.



Firefox Quantum

New. Fast. Fierce.

Launches November 14, 2017. Sign up to stay in the know and get launch updates.

MONDAYNOTES
Autonomous Cars: The Level 5 Fallacy
Last week, I "explained" how Google was set to become the

THEGUARDIAN
The edge of reason: the world's boldest climb and the man who conquered it
Climbers love history. Dates,

ECONOMIST
The lessons of fidget spinners
YOU can spin them on your nose, chin, finger or tongue. Some

How to ship quality?

moz://a

Quality?

- Three types of QA:
  - Catch issues during development phase
  - Automated tests & testsuites when the code land
  - Pre release channel (nightly, beta, etc)

Pre release testing

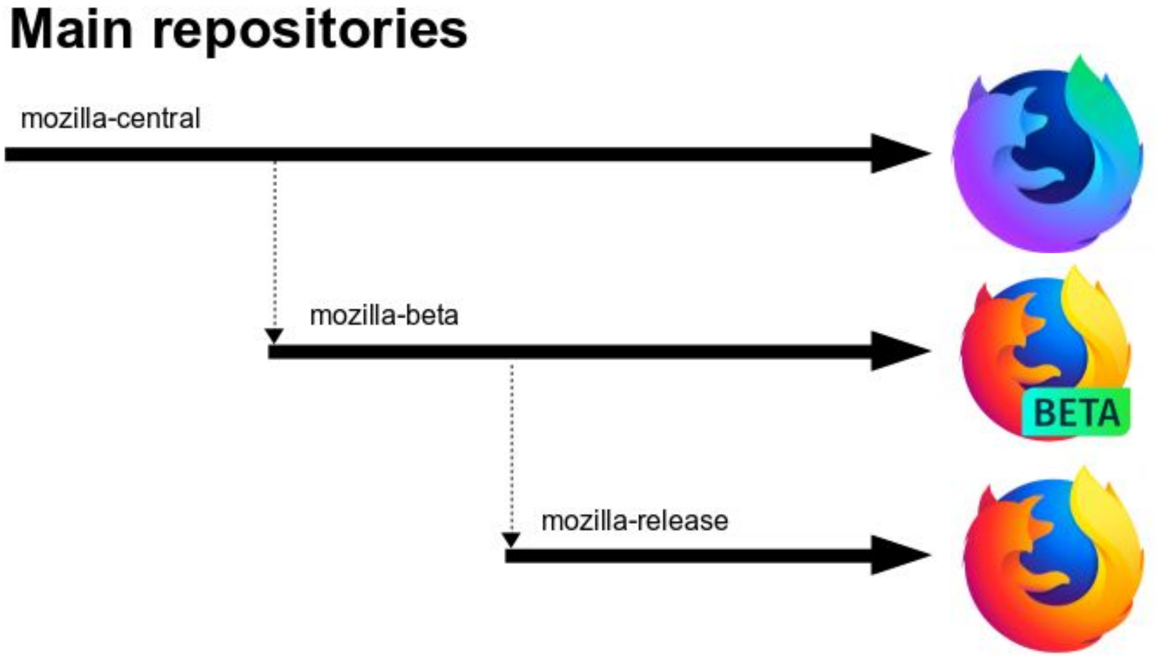
moz://a

Pre release testing

- The Web is a crazy platform
- All possible combinations of
  - HTML
  - CSS
  - Javascript (+ asm.js & WebAssembly)
  - Media format (Images, Audio, Video, etc)
  - Network
  - OS
  - ...

Pre release testing

- ➔ Release management - train model



Main repositories

Pre release testing

- We rely a lot on users on prerelease channel
  - Experiments (A/B testing) on pre-release channels

- Nightly - two nightlies per day
  - Hundred thousand of users

- Beta - 2 per week Desktop – 1 for Mobile
  - Millions of users

Manual testing

- Teams which test manually the new features

- Three colors
  - Green - Let’s ship it
  - Orange - We have to fix a few bugs
  - Red - Won’t be able to ship in this cycle

About:Firefox:Nightly

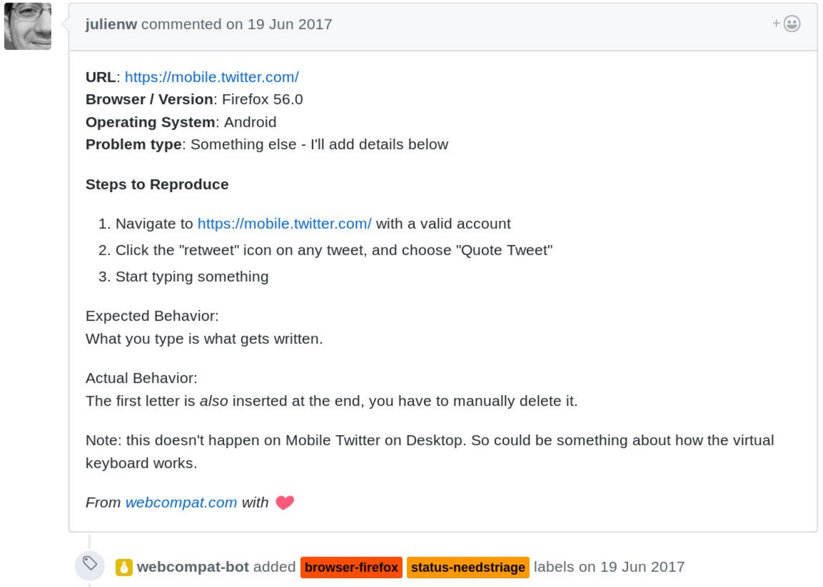
- Reboot of the nightly community
  - Paved the way for 57
  - Doubled the nightly population
  - 1184 bugs reported by the nightly community
  - @FirefoxNightly jumped from 9500 to 16600 followers

About:Firefox:Sumo

- ➔ Gather feedback from users
- ➔ Identify some hard issues like “my Firefox only shows blank pages”
- ➔ Share it to release management and other teams

Pre release testing - web compat

- ➔ Platform to report Web compatibility issues
- ➔ Different of behavior between browsers leading to rendering issues or JS errors



julienw commented on 19 Jun 2017

URL: https://mobile.twitter.com/
Browser / Version: Firefox 56.0
Operating System: Android
Problem type: Something else - I'll add details below

Steps to Reproduce

1. Navigate to https://mobile.twitter.com/ with a valid account
2. Click the "retweet" icon on any tweet, and choose "Quote Tweet"
3. Start typing something

Expected Behavior:
What you type is what gets written.

Actual Behavior:
The first letter is *also* inserted at the end, you have to manually delete it.

Note: this doesn't happen on Mobile Twitter on Desktop. So could be something about how the virtual keyboard works.

From webcompat.com with ❤

webcompat-bot added **browser-firefox** **status-needstriage** labels on 19 Jun 2017

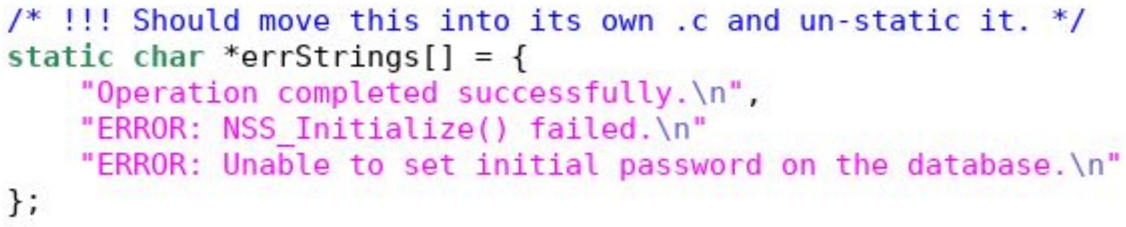
Code quality?

moz://a

Static analysis / linting

- ➔ C & C++ are hard languages like really really hard!
- ➔ How to detect programming mistakes
  - ◆ Related to the language designs
  - ◆ Usage of our APIs
  - ◆ Limit the code legacy

- ➔ Example:



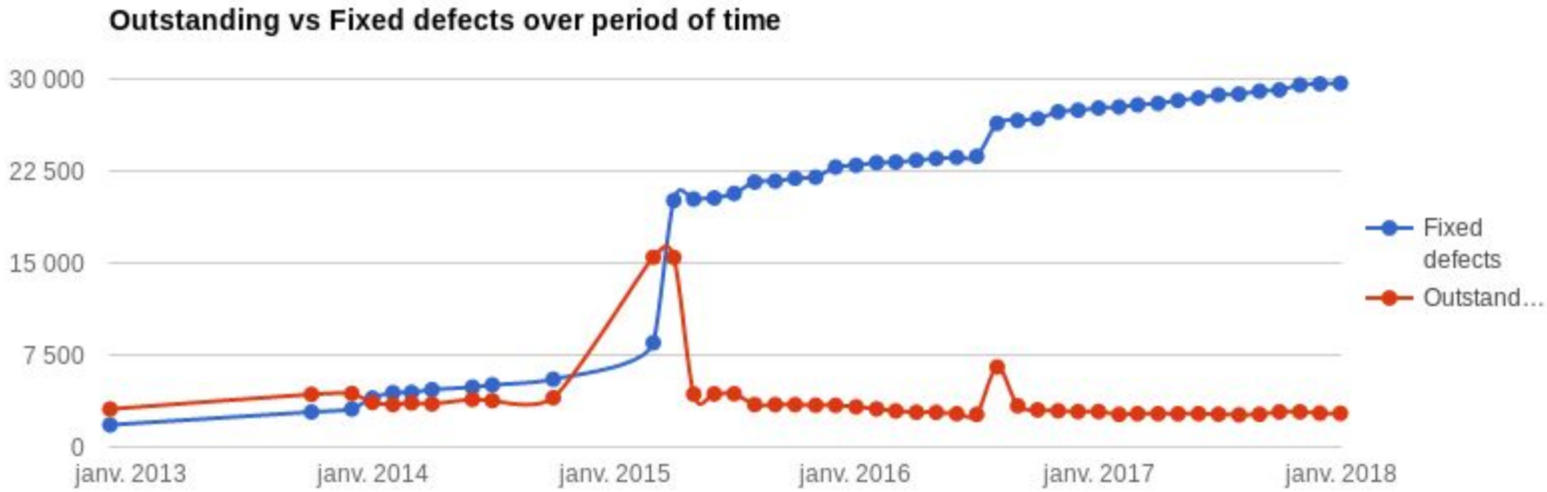
```
/* !!! Should move this into its own .c and un-static it. */
static char *errStrings[] = {
    "Operation completed successfully.\n",
    "ERROR: NSS_Initialize() failed.\n"
    "ERROR: Unable to set initial password on the database.\n"
};
```

Static analysis / linting

- Clang analyzer: 23 checkers
  - Dead code, insecure functions, etc
- Mozilla’s: 26 checkers
  - Security issues, bad usages of API, best practices
- clang-tidy : 28 checkers
  - Best practices, coding style, performances, C++ 11, 14 or 17 upgrade

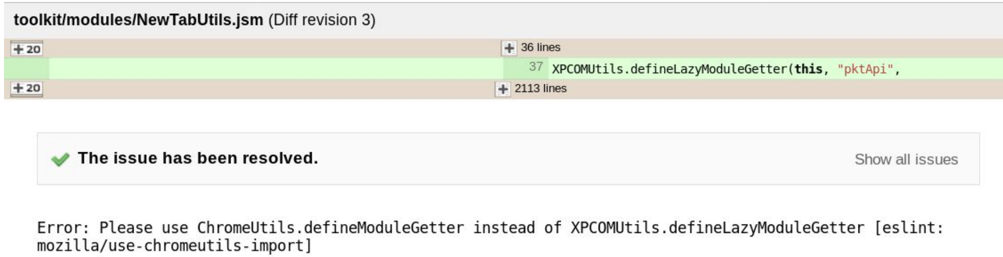
Static analysis / linting

- Once the code land, Coverity can catch others



SA tools that we use

- We use other tools for other languages
  - Javascript - Eslint
  - Python - flake8
  - Java (android) - findbug
  - Bash - shellcheck
  - Typos - codespell



toolkit/modules/NewTabUtils.jsm (Diff revision 3)

```
XPCOMUtils.defineLazyModuleGetter(this, "pktApi",
```

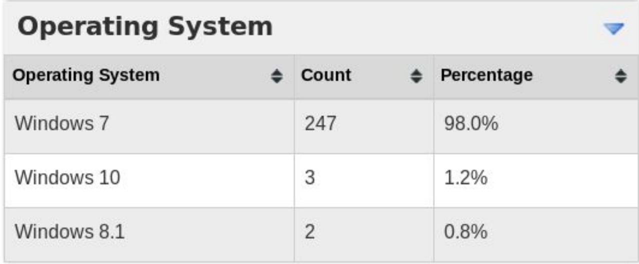
✓ **The issue has been resolved.**

```
Error: Please use ChromeUtils.defineModuleGetter instead of XPCOMUtils.defineLazyModuleGetter [eslint:
mozilla/use-chromeutils-import]
```

- For every commit – average of 12 minutes analysis
- We contribute upstream and sponsor some projects (ex: LLVM)...

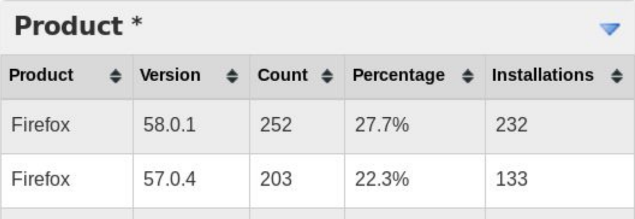
Crash analysis

- When a crash occurs
  - Handled by breakpad
  - Sent to https://crash-stats.mozilla.com/
  - Doing some voodoo magic on them



Operating System

Operating System	Count	Percentage
Windows 7	247	98.0%
Windows 10	3	1.2%
Windows 8.1	2	0.8%

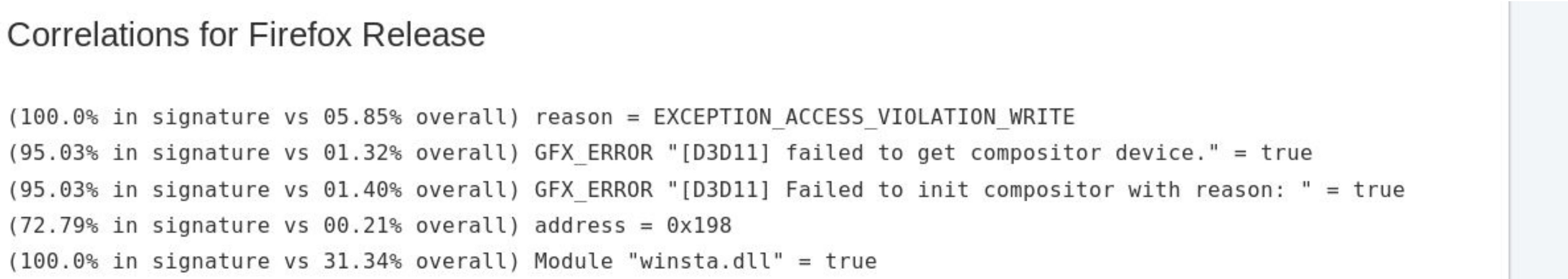


Product *

Product	Version	Count	Percentage	Installations
Firefox	58.0.1	252	27.7%	232
Firefox	57.0.4	203	22.3%	133

Crash analysis

- ➔ Data mining on the results

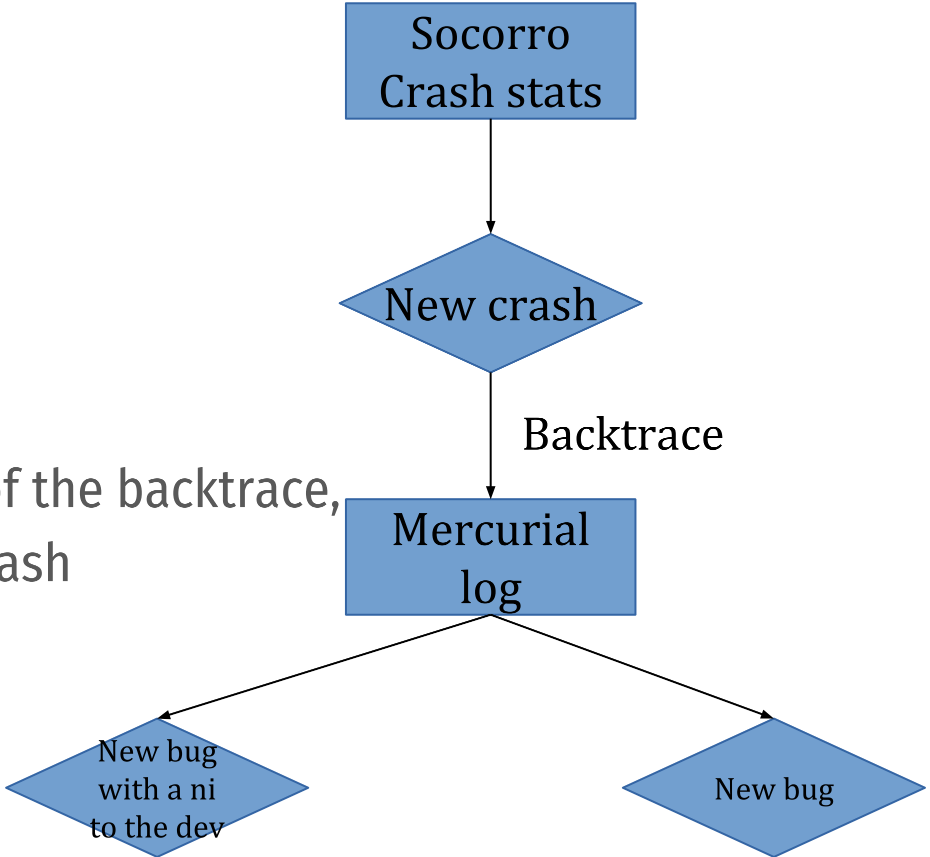


Correlations for Firefox Release

```
(100.0% in signature vs 05.85% overall) reason = EXCEPTION_ACCESS_VIOLATION_WRITE
(95.03% in signature vs 01.32% overall) GFX_ERROR "[D3D11] failed to get compositor device." = true
(95.03% in signature vs 01.40% overall) GFX_ERROR "[D3D11] Failed to init compositor with reason: " = true
(72.79% in signature vs 00.21% overall) address = 0x198
(100.0% in signature vs 31.34% overall) Module "winsta.dll" = true
```

Crash analysis - clouseau

- ➔ Look at new crash signatures
- ➔ Extract the backtrace
- ➔ Look at the recent VCS history
- ➔ If a change touched one level of the backtrace, it might be the source of the crash
- ➔ 212 bugs reported



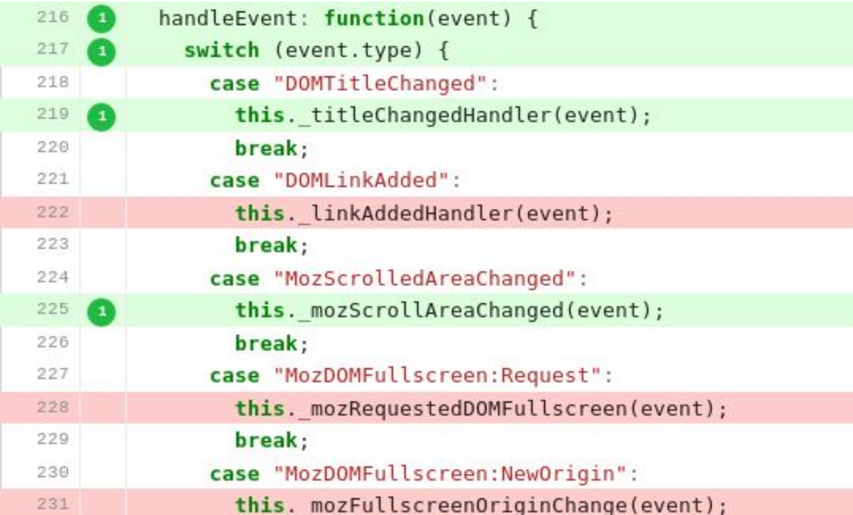
Code coverage

- Understanding of the quality of the testsuites

- (afaik) First time done on this scale
- We had to:
  - Add JS code coverage support in the Firefox JS engine
  - Add code coverage support in the Rust compiler
  - Patch: gcc, llvm, clang & compiler-rt
  - Develop an alternative to lcov called grcov dropping the processing time from more than 24 hours to less than 5 minutes

Code coverage - the results

- Current code coverage results (Windows & Linux)
  - C++
    - 2 913 824 lines
    - 1 620 227 covered lines (55,6%)
  - JS
    - 586 383 lines
    - 426 906 covered lines (72,8%)



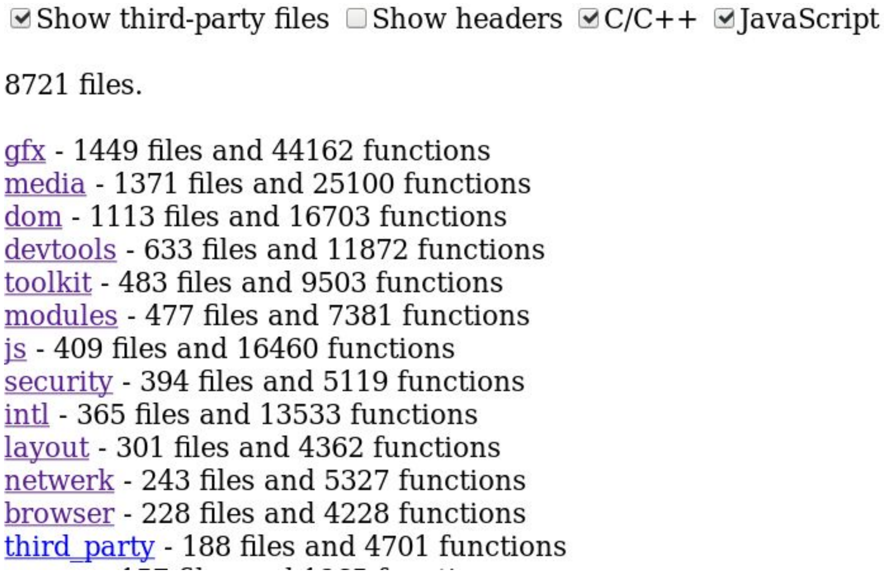
```
handleEvent: function(event) {
  switch (event.type) {
    case "DOMTitleChanged":
      this._titleChangedHandler(event);
      break;
    case "DOMLinkAdded":
      this._linkAddedHandler(event);
      break;
    case "MozScrolledAreaChanged":
      this._mozScrollAreaChanged(event);
      break;
    case "MozDOMFullscreen:Request":
      this._mozRequestedDOMFullscreen(event);
      break;
    case "MozDOMFullscreen:NewOrigin":
      this._mozFullscreenOriginChange(event);
```

Code coverage - Side effect

- ➔ What does it mean when a file has 0 coverage?

  A bug, for sure!

  - ◆ Dead code
  - ◆ No test



☑ Show third-party files ☐ Show headers ☑ C/C++ ☑ JavaScript

8721 files.

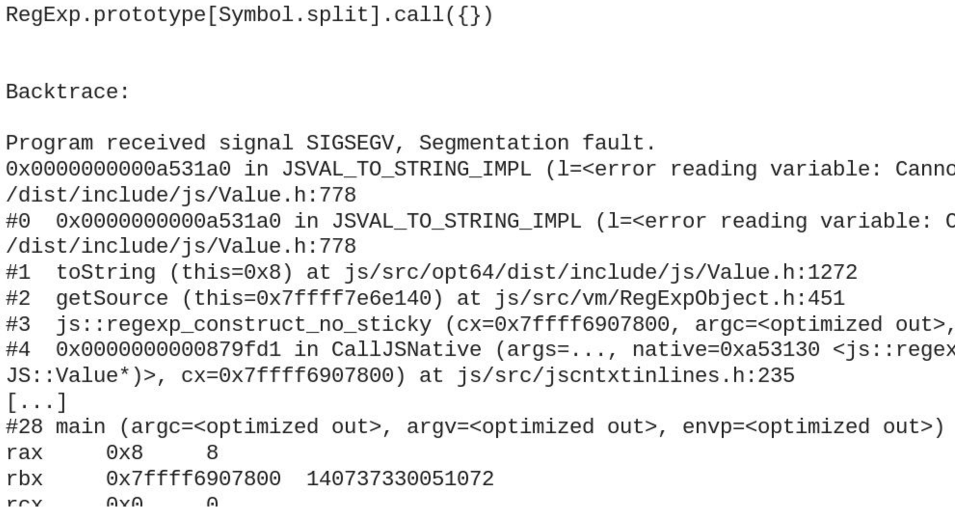
gfx - 1449 files and 44162 functions
media - 1371 files and 25100 functions
dom - 1113 files and 16703 functions
devtools - 633 files and 11872 functions
toolkit - 483 files and 9503 functions
modules - 477 files and 7381 functions
js - 409 files and 16460 functions
security - 394 files and 5119 functions
intl - 365 files and 13533 functions
layout - 301 files and 4362 functions
netwerk - 243 files and 5327 functions
browser - 228 files and 4228 functions
third_party - 188 files and 4701 functions

- ➔ 61 removed files

  13272 removed lines.

Fuzzing

- Send invalid, unexpected, or random data as inputs
- We are testing:
  - JavaScript features, DOM, Layout, CSS, Stylo,
  - Media file formats (images, audio, video)
- Last 2 y, over 600 security bugs



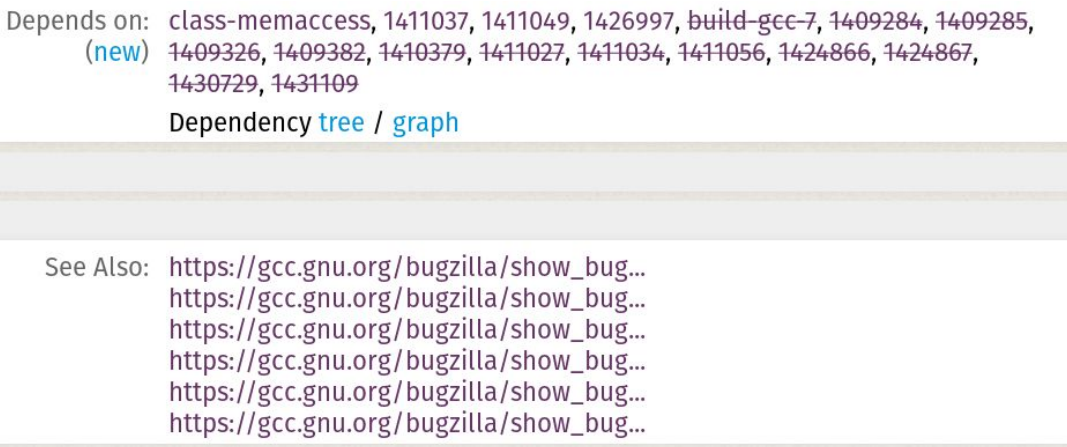
```
RegExp.prototype[Symbol.split].call({})


Backtrace:

Program received signal SIGSEGV, Segmentation fault.
0x0000000000a531a0 in JSVAL_TO_STRING_IMPL (l=<error reading variable: Canno
/dist/include/js/Value.h:778
#0  0x0000000000a531a0 in JSVAL_TO_STRING_IMPL (l=<error reading variable: C
/dist/include/js/Value.h:778
#1  toString (this=0x8) at js/src/opt64/dist/include/js/Value.h:1272
#2  getSource (this=0x7ffff7e6e140) at js/src/vm/RegExpObject.h:451
#3  js::regexp_construct_no_sticky (cx=0x7ffff6907800, argc=<optimized out>,
#4  0x0000000000879fd1 in CallJSNative (args=..., native=0xa53130 <js::regex
JS::Value*)>, cx=0x7ffff6907800) at js/src/jscntxtinlines.h:235
[...]
#28 main (argc=<optimized out>, argv=<optimized out>, envp=<optimized out>)
rax     0x8      8
rbx     0x7ffff6907800  140737330051072
rcx     0x0      0
```

Other best practices

- ➔ Once or twice a day, compiler Firefox trunk with -Werror on:
  - ◆ Build with gcc snapshot packages from Debian experimental (currently version 8)
  - ◆ Clang trunk (currently version 7)

- ➔ Find new issues in our code

- ➔ Find bugs in the compiler



Depends on: (new) class-memaccess, 1411037, 1411049, 1426997, ~~build-gcc-7~~, ~~1409284~~, ~~1409285~~, ~~1409326~~, ~~1409382~~, ~~1410379~~, ~~1411027~~, ~~1411034~~, ~~1411056~~, ~~1424866~~, ~~1424867~~, ~~1430729~~, ~~1431109~~

Dependency tree / graph

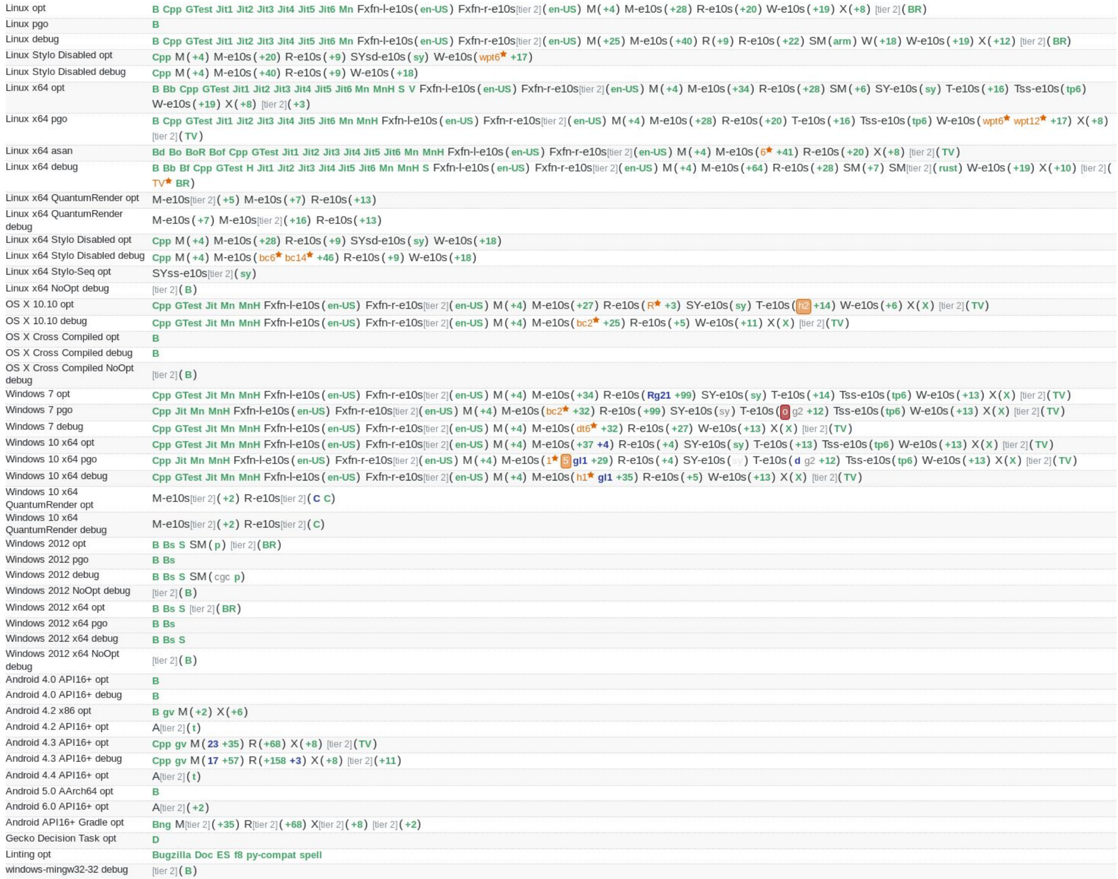
See Also: https://gcc.gnu.org/bugzilla/show_bug...
https://gcc.gnu.org/bugzilla/show_bug...
https://gcc.gnu.org/bugzilla/show_bug...
https://gcc.gnu.org/bugzilla/show_bug...
https://gcc.gnu.org/bugzilla/show_bug...
https://gcc.gnu.org/bugzilla/show_bug...

moz://a

Automation

moz://a

Crazy CI



Linux opt	B Cpp GTest Jit1 Jit2 Jit3 Jit4 Jit5 Jit6 Mn Fxfn-l-e10s (en-US) Fxfn-r-e10s[tier 2] (en-US) M (+4) M-e10s (+28) R-e10s (+20) W-e10s (+19) X (+8) [tier 2] (BR)
Linux pgo	B
Linux debug	B Cpp GTest Jit1 Jit2 Jit3 Jit4 Jit5 Jit6 Mn Fxfn-l-e10s (en-US) Fxfn-r-e10s[tier 2] (en-US) M (+25) M-e10s (+40) R (+9) R-e10s (+22) SM (arm) W (+18) W-e10s (+19) X (+12) [tier 2] (BR)
Linux Stylo Disabled opt	Cpp M (+4) M-e10s (+20) R-e10s (+9) SYsd-e10s (sy) W-e10s (wpt6★ +17)
Linux Stylo Disabled debug	Cpp M (+4) M-e10s (+40) R-e10s (+9) W-e10s (+18)
Linux x64 opt	B Bb Cpp GTest Jit1 Jit2 Jit3 Jit4 Jit5 Jit6 Mn MnH S V Fxfn-l-e10s (en-US) Fxfn-r-e10s[tier 2] (en-US) M (+4) M-e10s (+34) R-e10s (+28) SM (+6) SY-e10s (sy) T-e10s (+16) Tss-e10s (tp6) W-e10s (+19) X (+8) [tier 2] (+3)
Linux x64 pgo	B Cpp GTest Jit1 Jit2 Jit3 Jit4 Jit5 Jit6 Mn MnH Fxfn-l-e10s (en-US) Fxfn-r-e10s[tier 2] (en-US) M (+4) M-e10s (+28) R-e10s (+20) T-e10s (+16) Tss-e10s (tp6) W-e10s (wpt6★ wpt12★ +17) X (+8) [tier 2] (TV)
Linux x64 asan	Bd Bo BoR Bof Cpp GTest Jit1 Jit2 Jit3 Jit4 Jit5 Jit6 Mn MnH Fxfn-l-e10s (en-US) Fxfn-r-e10s[tier 2] (en-US) M (+4) M-e10s (6★ +41) R-e10s (+20) X (+8) [tier 2] (TV)
Linux x64 debug	B Bb Bf Cpp GTest H Jit1 Jit2 Jit3 Jit4 Jit5 Jit6 Mn MnH S Fxfn-l-e10s (en-US) Fxfn-r-e10s[tier 2] (en-US) M (+4) M-e10s (+64) R-e10s (+28) SM (+7) SM[tier 2] (rust) W-e10s (+19) X (+10) [tier 2] (TV★ BR)
Linux x64 QuantumRender opt	M-e10s[tier 2] (+5) M-e10s (+7) R-e10s (+13)
Linux x64 QuantumRender debug	M-e10s (+7) M-e10s[tier 2] (+16) R-e10s (+13)
Linux x64 Stylo Disabled opt	Cpp M (+4) M-e10s (+28) R-e10s (+9) SYsd-e10s (sy) W-e10s (+18)
Linux x64 Stylo Disabled debug	Cpp M (+4) M-e10s (bc6★ bc14★ +46) R-e10s (+9) W-e10s (+18)
Linux x64 Stylo-Seq opt	SYss-e10s[tier 2] (sy)
Linux x64 NoOpt debug	[tier 2] (B)
OS X 10.10 opt	Cpp GTest Jit Mn MnH Fxfn-l-e10s (en-US) Fxfn-r-e10s[tier 2] (en-US) M (+4) M-e10s (+27) R-e10s (R★ +3) SY-e10s (sy) T-e10s (h2 +14) W-e10s (+6) X (X) [tier 2] (TV)
OS X 10.10 debug	Cpp GTest Jit Mn MnH Fxfn-l-e10s (en-US) Fxfn-r-e10s[tier 2] (en-US) M (+4) M-e10s (bc2★ +25) R-e10s (+5) W-e10s (+11) X (X) [tier 2] (TV)
OS X Cross Compiled opt	B
OS X Cross Compiled debug	B
OS X Cross Compiled NoOpt debug	[tier 2] (B)
Windows 7 opt	Cpp GTest Jit Mn MnH Fxfn-l-e10s (en-US) Fxfn-r-e10s[tier 2] (en-US) M (+4) M-e10s (+34) R-e10s (Rg21 +99) SY-e10s (sy) T-e10s (+14) Tss-e10s (tp6) W-e10s (+13) X (X) [tier 2] (TV)
Windows 7 pgo	Cpp Jit Mn MnH Fxfn-l-e10s (en-US) Fxfn-r-e10s[tier 2] (en-US) M (+4) M-e10s (bc2★ +32) R-e10s (+99) SY-e10s (sy) T-e10s (o g2 +12) Tss-e10s (tp6) W-e10s (+13) X (X) [tier 2] (TV)
Windows 7 debug	Cpp GTest Jit Mn MnH Fxfn-l-e10s (en-US) Fxfn-r-e10s[tier 2] (en-US) M (+4) M-e10s (dt6★ +32) R-e10s (+27) W-e10s (+13) X (X) [tier 2] (TV)
Windows 10 x64 opt	Cpp GTest Jit Mn MnH Fxfn-l-e10s (en-US) Fxfn-r-e10s[tier 2] (en-US) M (+4) M-e10s (+37 +4) R-e10s (+4) SY-e10s (sy) T-e10s (+13) Tss-e10s (tp6) W-e10s (+13) X (X) [tier 2] (TV)
Windows 10 x64 pgo	Cpp Jit Mn MnH Fxfn-l-e10s (en-US) Fxfn-r-e10s[tier 2] (en-US) M (+4) M-e10s (1★ 5 gl1 +29) R-e10s (+4) SY-e10s (sy) T-e10s (d g2 +12) Tss-e10s (tp6) W-e10s (+13) X (X) [tier 2] (TV)
Windows 10 x64 debug	Cpp GTest Jit Mn MnH Fxfn-l-e10s (en-US) Fxfn-r-e10s[tier 2] (en-US) M (+4) M-e10s (h1★ gl1 +35) R-e10s (+5) W-e10s (+13) X (X) [tier 2] (TV)
Windows 10 x64 QuantumRender opt	M-e10s[tier 2] (+2) R-e10s[tier 2] (C C)
Windows 10 x64 QuantumRender debug	M-e10s[tier 2] (+2) R-e10s[tier 2] (C)
Windows 2012 opt	B Bs S SM (p) [tier 2] (BR)
Windows 2012 pgo	B Bs
Windows 2012 debug	B Bs S SM (cgc p)
Windows 2012 NoOpt debug	[tier 2] (B)
Windows 2012 x64 opt	B Bs S [tier 2] (BR)
Windows 2012 x64 pgo	B Bs
Windows 2012 x64 debug	B Bs S
Windows 2012 x64 NoOpt debug	[tier 2] (B)
Android 4.0 API16+ opt	B
Android 4.0 API16+ debug	B
Android 4.2 x86 opt	B gv M (+2) X (+6)
Android 4.2 API16+ opt	A[tier 2] (t)
Android 4.3 API16+ opt	Cpp gv M (23 +35) R (+68) X (+8) [tier 2] (TV)
Android 4.3 API16+ debug	Cpp gv M (17 +57) R (+158 +3) X (+8) [tier 2] (+11)
Android 4.4 API16+ opt	A[tier 2] (t)
Android 5.0 AArch64 opt	B
Android 6.0 API16+ opt	A[tier 2] (+2)
Android API16+ Gradle opt	Bng M[tier 2] (+35) R[tier 2] (+68) X[tier 2] (+8) [tier 2] (+2)
Gecko Decision Task opt	D
Linting opt	Bugzilla Doc ES f8 py-compat spell
windows-mingw32-32 debug	[tier 2] (B)

CI

- Launched (almost) on every commit

- Can be used by individual developers
  - Platforms
  - With testsuite selection

WPT - Web Platform Tests

- A W3C-coordinated attempt to build a cross-browser testsuite for the Web-platform stack

- Takes ~184 minutes on a Linux PGO build

Despite all that

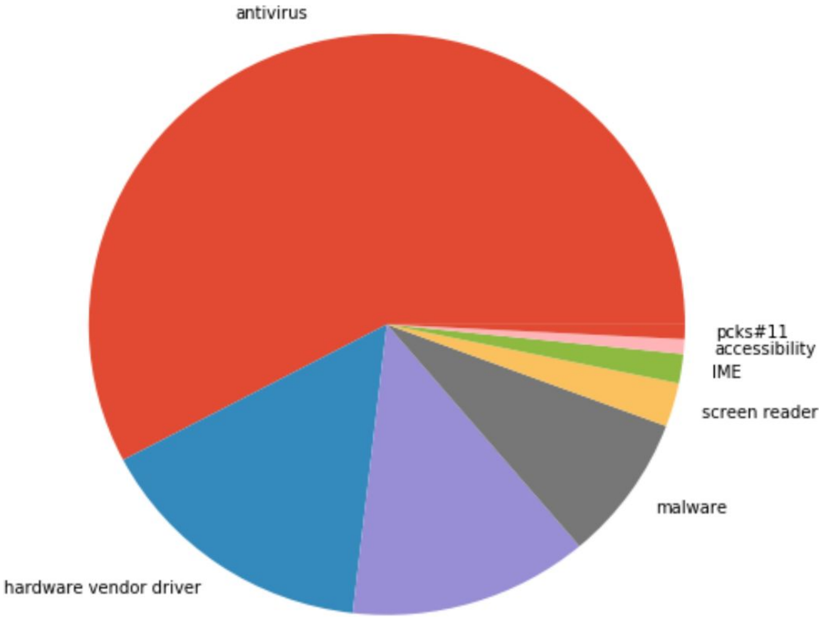
moz://a

Predicting release quality is tough…….

- ➔ We still have issues after we published to users

- ➔ Huge trunk caused by Malware, Antivirus or security software

- ➔ Some web compat issues
(example: Outlook web in 58)



moz://a

How to contribute

How to contribute?

- ➔ Don't be shy. We have all been beginners at some point
- ➔ RTFM
- ➔ Find a mentor/friend to guide you
- ➔ Use Firefox nightly/Rust trunk

Report bugs - https://bugzilla.mozilla.org/

How to contribute? - Think small

- ➔ Try to build Firefox
- ➔ Start with a small bug
- ➔ Easy example : remove a C/C++ warning, typo in the code, etc
    - ◆ Don't try to fix a big issue
    - ◆ You will probably fail
- ➔ Workflows represent an important part of the contribution process

How to contribute? -

- ➔ https://whatcanidoformozilla.org
- ➔ http://www.joshmatthews.net/bugsahoy/
- ➔ http://sylvestre.ledru.info/reports/fx-scan-build/report-nsFrame.cpp-BuildDisplayListForChild-251-1.html#EndPath

moz://a