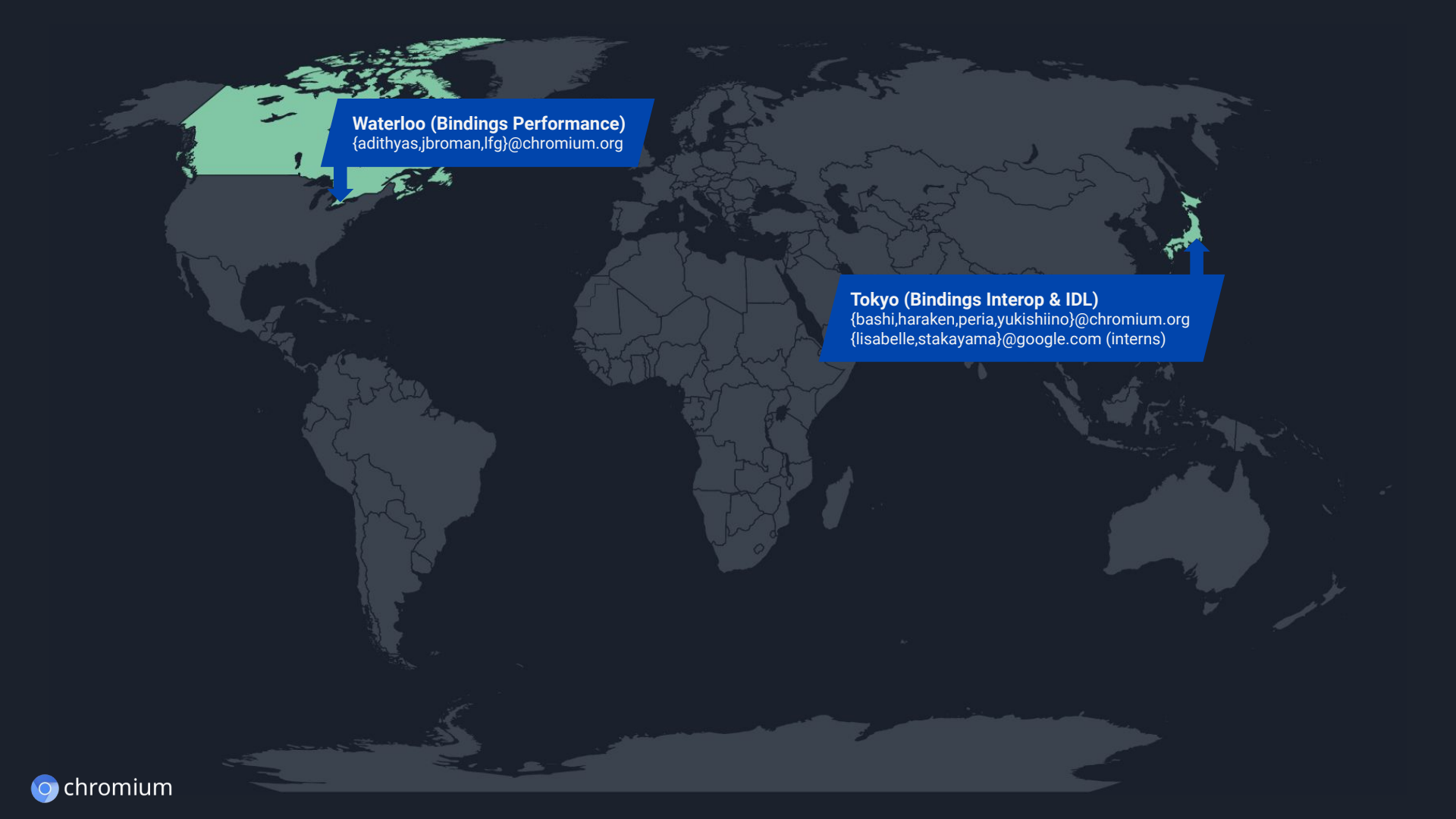
Speedometer Update

jbroman@chromium.org
Blink Bindings Team
BlinkOn 8 – Tokyo

chromium


Waterloo (Bindings Performance)
{adithyas,jbroman,lfg}@chromium.org
Tokyo (Bindings Interop & IDL)
{bashi,haraken,peria,yukishiino}@chromium.org
{lisabelle,stakayama}@google.com (interns)

chromium

What is speedometer ?

Speedometer

Benchmark	Vendor	Released
SunSpider	Apple	2007
Dromaeo	Mozilla	2008
Kraken	Mozilla	2010
Octane	Google	2012
Speedometer	Apple	2014
JetStream	Apple	2014
ARES-6	Apple	2017

Speedometer: Benchmark for Web App Responsiveness

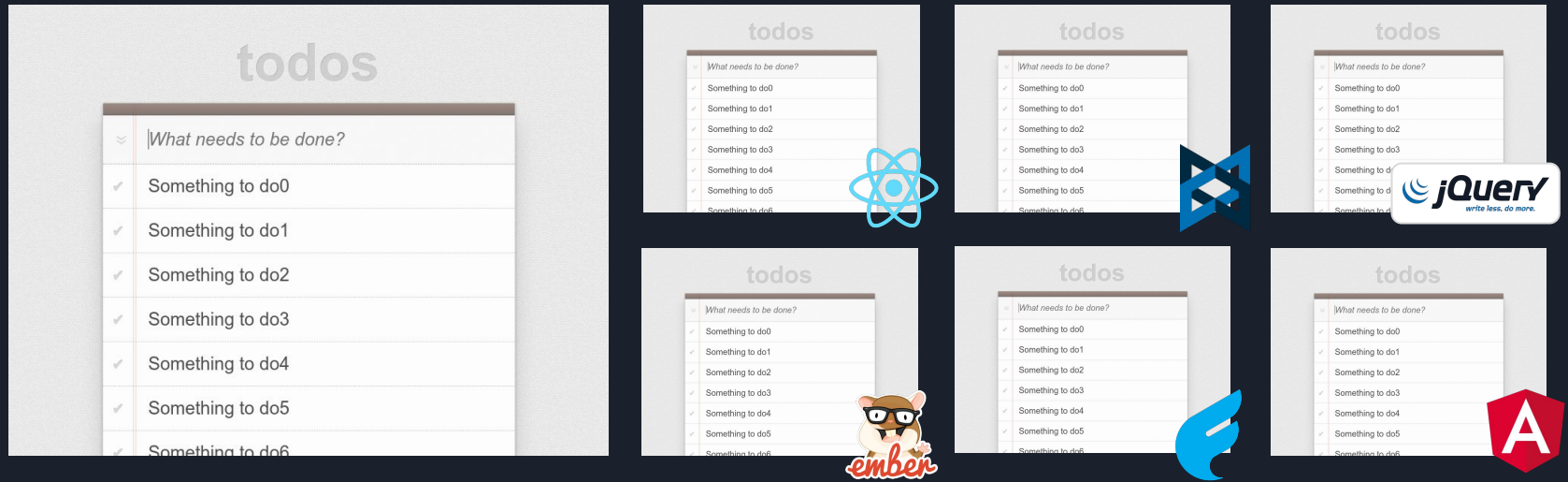
Jun 2, 2014
by Ryosuke Niwa

Today we are pleased to announce Speedometer, a new benchmark that measures the responsiveness of web applications.

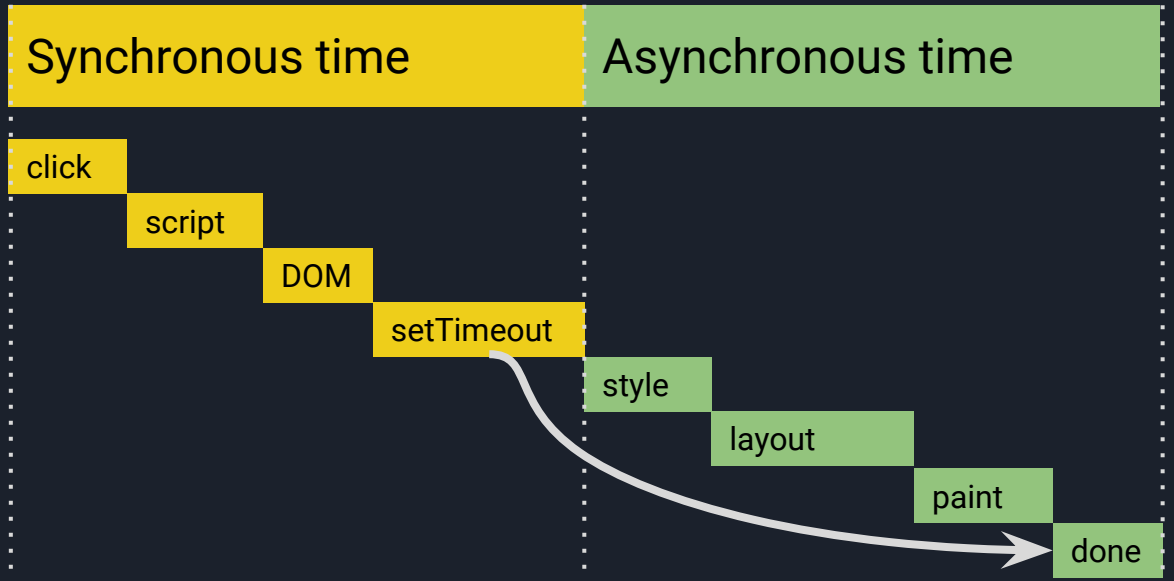
Benchmark

Speedometer measures simulated user interactions in web applications. Version 1.0 of Speedometer uses TodoMVC to simulate user actions for adding, completing, and removing to-do items. Speedometer repeats the same actions using DOM APIs — a core set of web platform APIs used extensively in web applications — as well as six popular JavaScript frameworks: Ember.js, Backbone.js, jQuery, AngularJS, React, and Flight. Many of these frameworks are used on the most popular websites in the world, such as Facebook and Twitter. The performance of these types of operations depends on the speed of the DOM APIs, the JavaScript engine, CSS style resolution, layout, and other technologies.

Speedometer: TodoMVC ×7



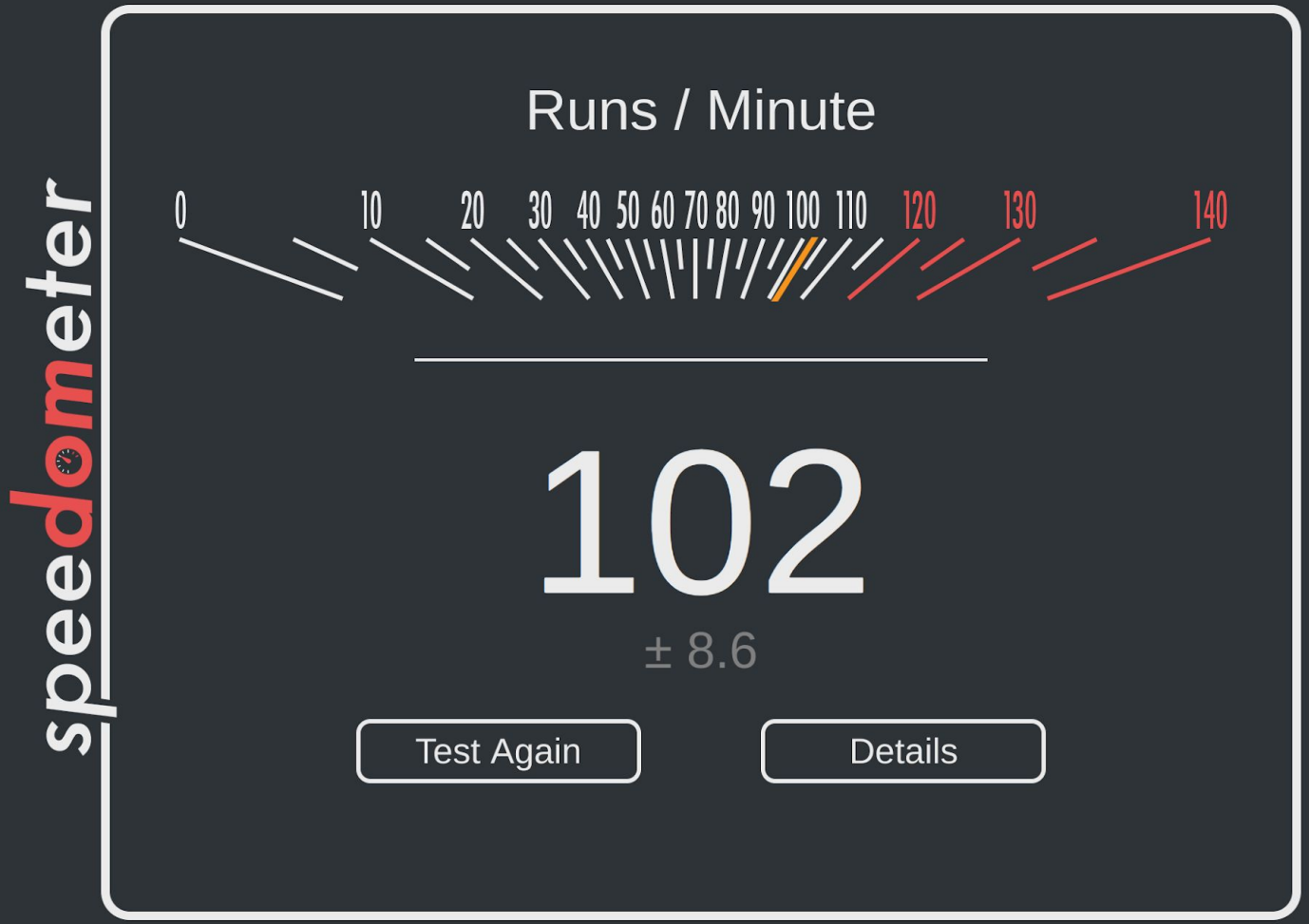
Speedometer: Timing



Speedometer: Scoring

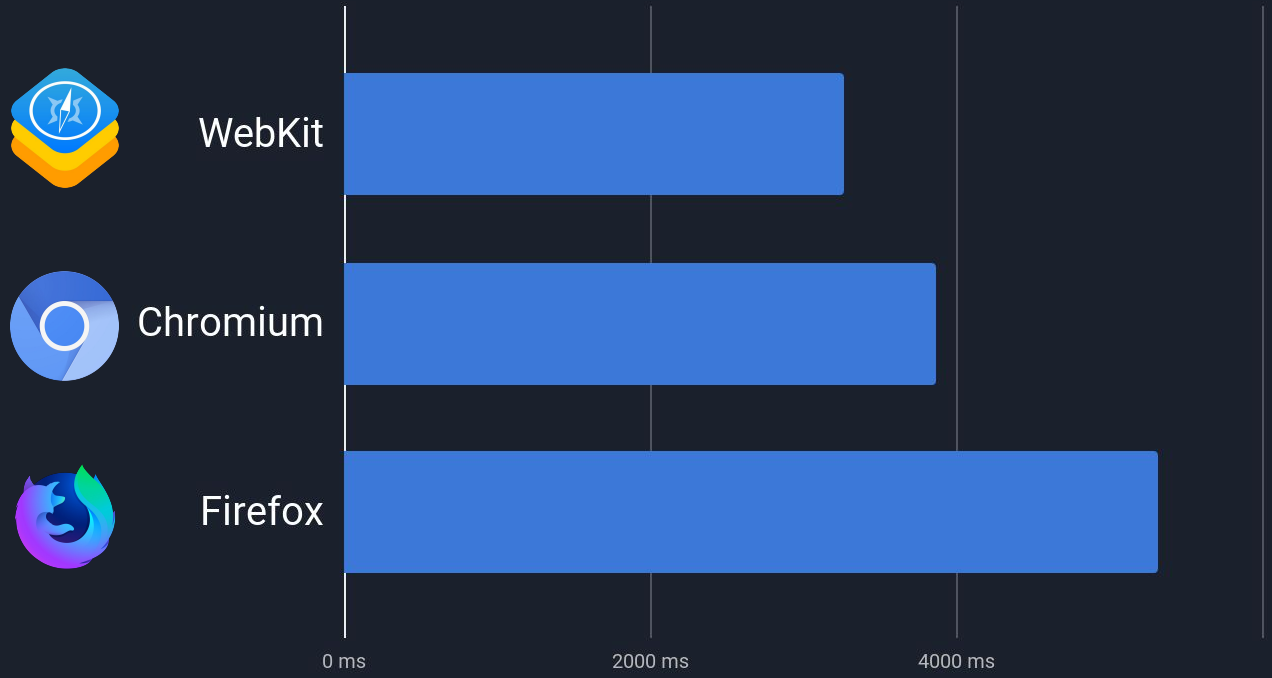
$$\text{overall score} = \frac{1}{\sum_i (\text{sync time}_i + \text{async time}_i)}$$

- The standard runner displays this in units of runs/min
- Inversely proportional to the (arithmetic) mean time to execute each test
- Slower tests (e.g. EmberJS-TodoMVC) weighted more heavily
- Speedometer 2.0 switches to a geometric mean


speedometer
Runs / Minute
0
10
20
30
40
50
60
70
80
90
100
110
120
130
140
102
± 8.6
Test Again
Details

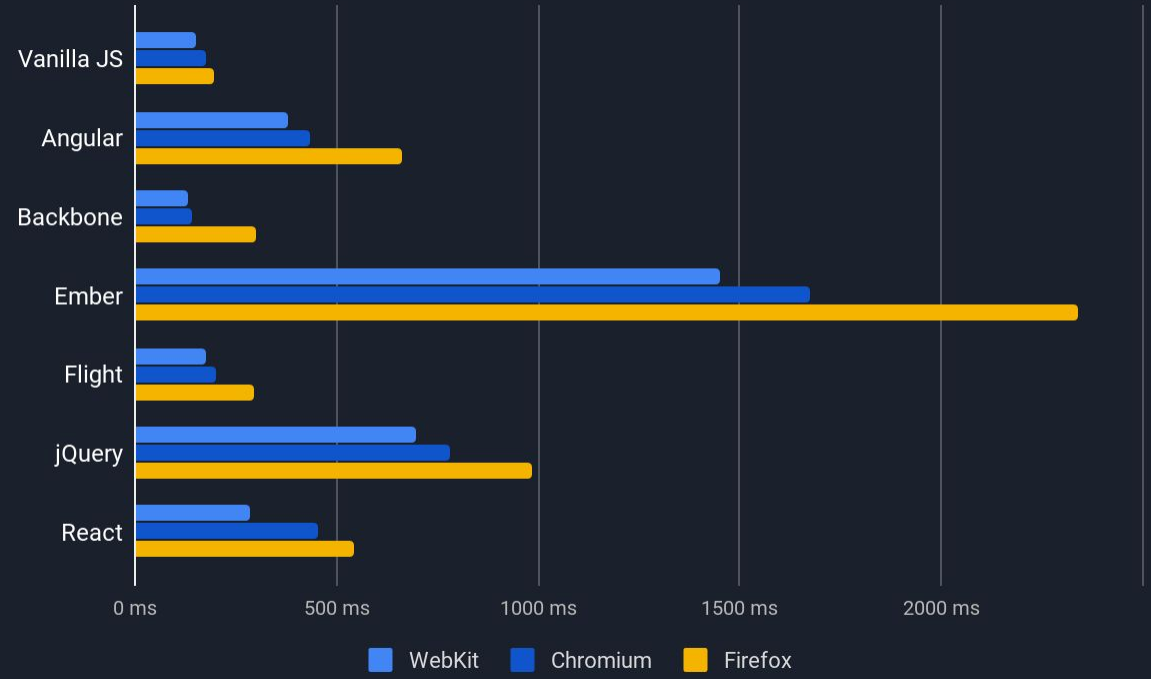
How are we doing?

Speedometer: Overall



Time in ms, lower is better
Apple Macbook Pro (macOS)
WebKit Nightly, Chrome Canary, Firefox Nightly

Speedometer: By Framework

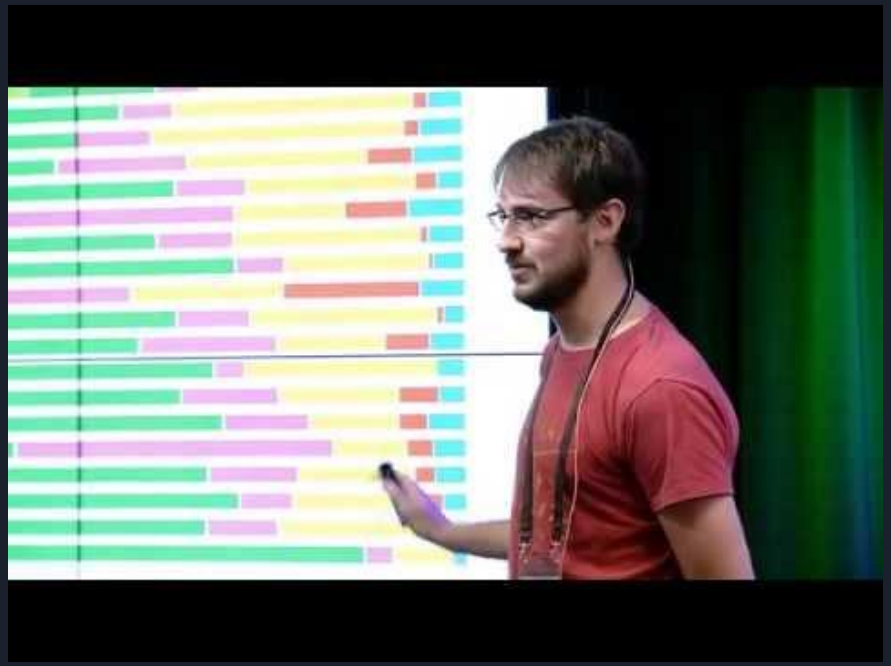


Time in ms, lower is better
Apple Macbook Pro (macOS)
WebKit Nightly, Chrome Canary, Firefox Nightly

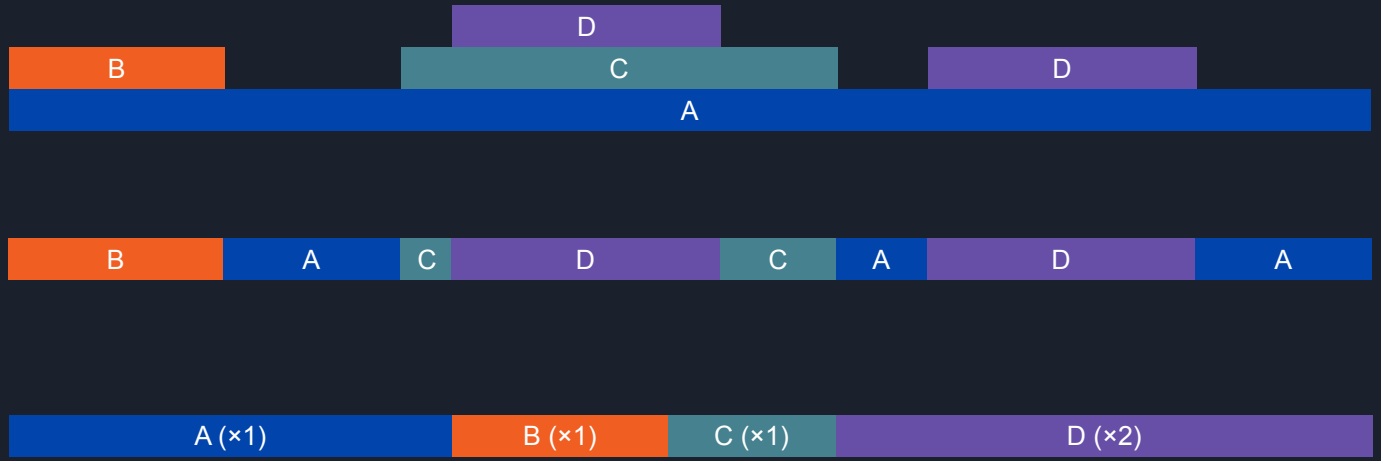
chromium

Where are we spending our time?

V8 RuntimeCallStats



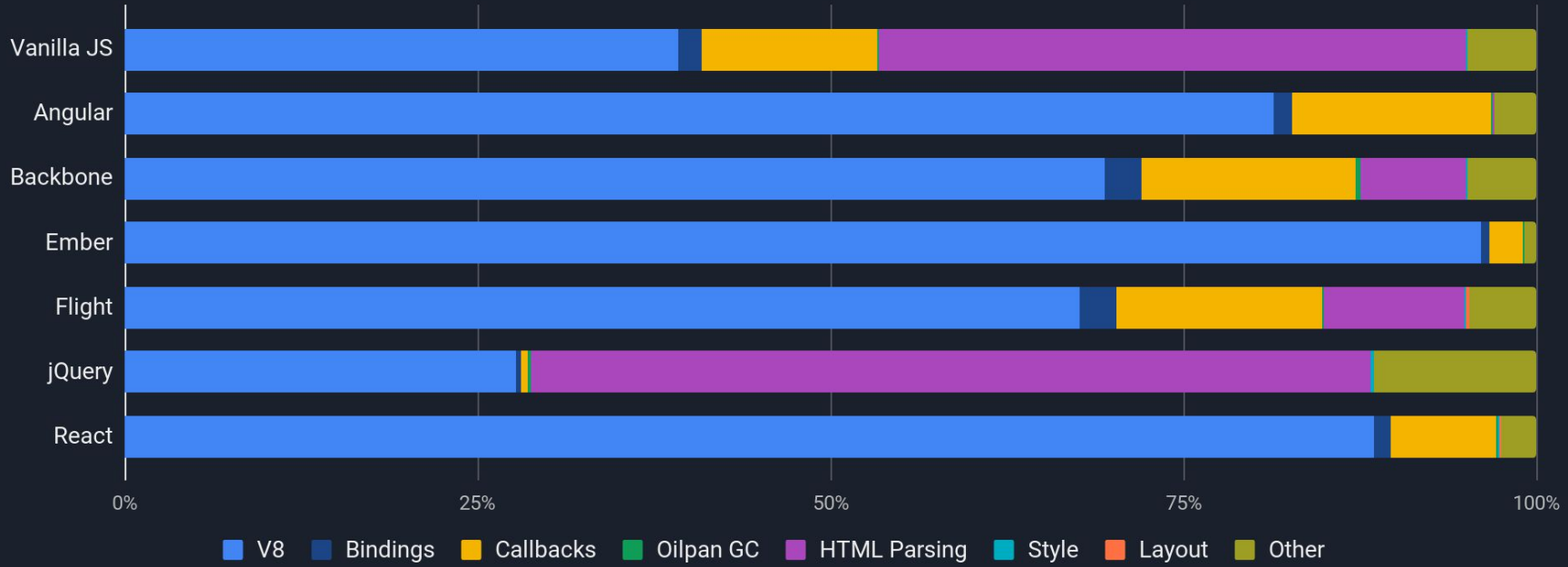
V8 RuntimeCallStats



Blink RuntimeCallStats

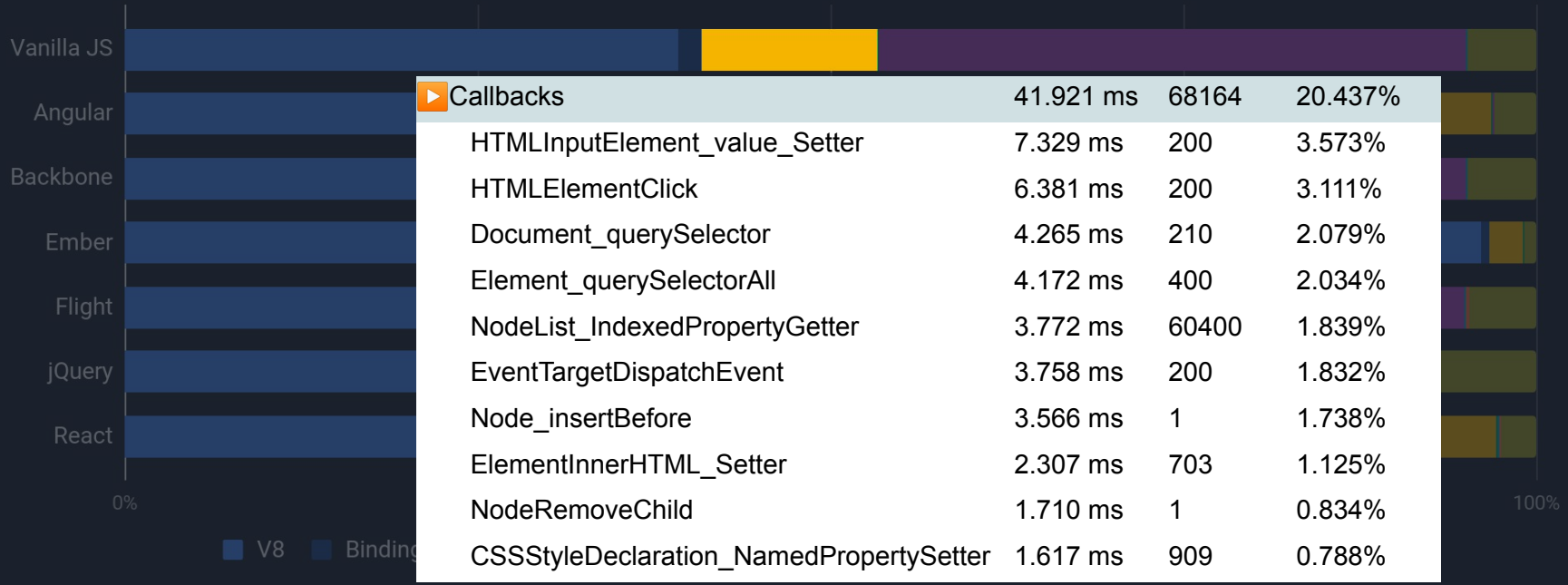
- Similar to V8 RuntimeCallStats, aiming to explain “Blink C++”
- Counters on various high-level Blink subsystems (e.g. layout) and on specific bindings code (e.g. wrapper object creation)
- For detailed analysis, compile-time flag enables counters at all Blink bindings entry points
- Runs *during script execution*; does not presently capture time spent when script is not executing (e.g. frame-driven rendering)

Blink RCS & Speedometer



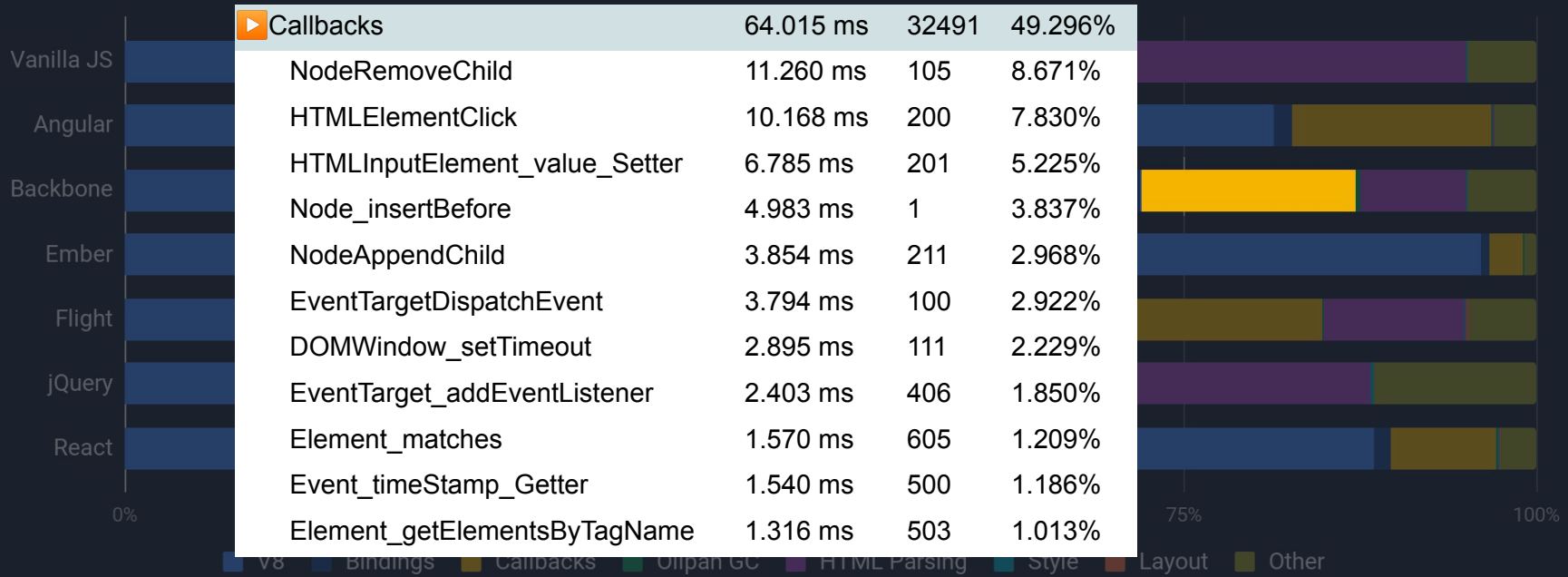
chromium

Blink RCS & Speedometer



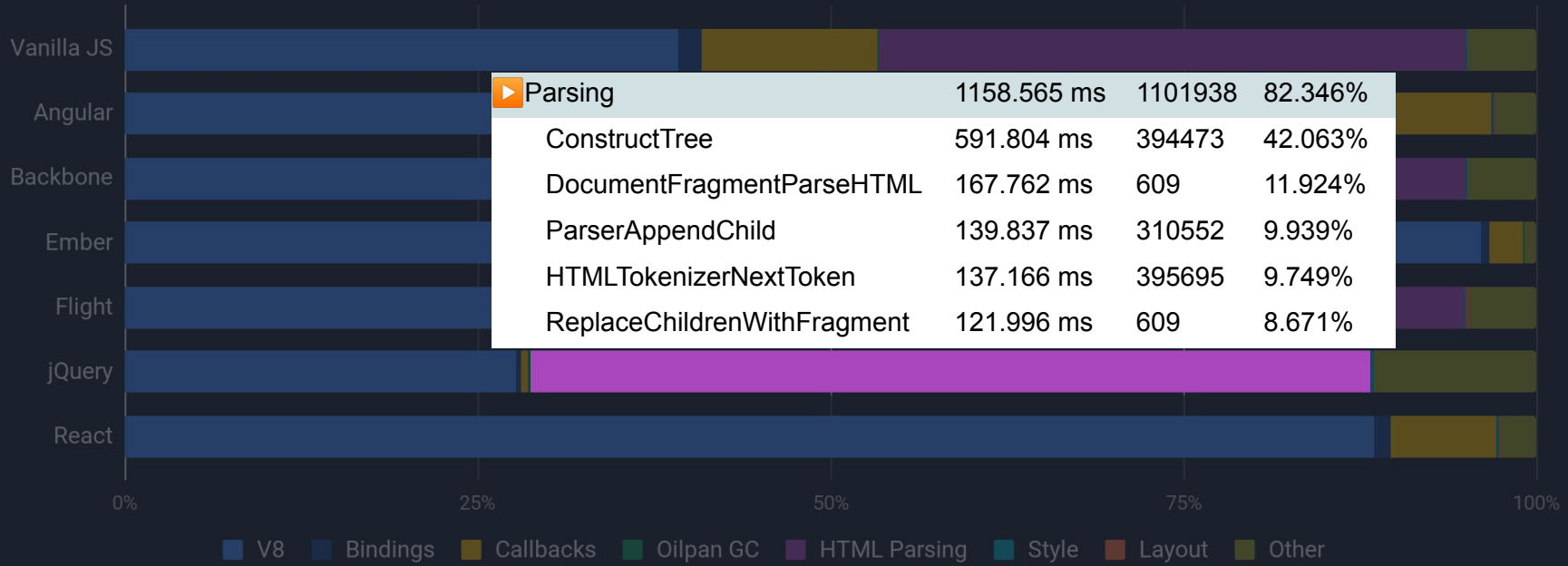
Callbacks	41.921 ms	68164	20.437%
HTMLInputElement_value_Setter	7.329 ms	200	3.573%
HTMLElementClick	6.381 ms	200	3.111%
Document_querySelector	4.265 ms	210	2.079%
Element_querySelectorAll	4.172 ms	400	2.034%
NodeList_IndexedPropertyGetter	3.772 ms	60400	1.839%
EventTargetDispatchEvent	3.758 ms	200	1.832%
Node_insertBefore	3.566 ms	1	1.738%
ElementInnerHTML_Setter	2.307 ms	703	1.125%
NodeRemoveChild	1.710 ms	1	0.834%
CSSStyleDeclaration_NamedPropertySetter	1.617 ms	909	0.788%

Blink RCS & Speedometer

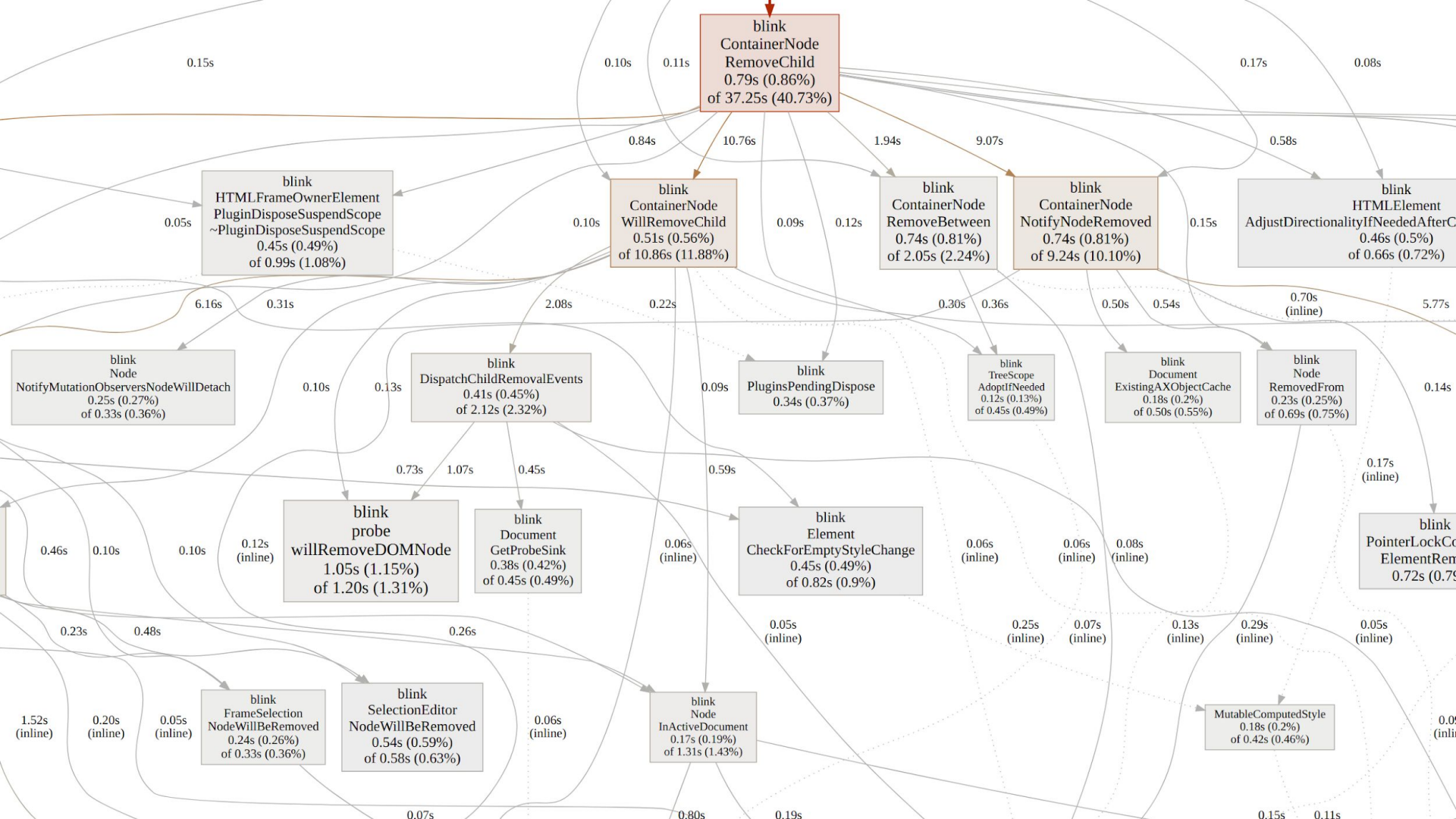


Callbacks	64.015 ms	32491	49.296%
NodeRemoveChild	11.260 ms	105	8.671%
HTMLElementClick	10.168 ms	200	7.830%
HTMLInputElement_value_Setter	6.785 ms	201	5.225%
Node_insertBefore	4.983 ms	1	3.837%
NodeAppendChild	3.854 ms	211	2.968%
EventTargetDispatchEvent	3.794 ms	100	2.922%
DOMWindow_setTimeout	2.895 ms	111	2.229%
EventTarget_addEventListener	2.403 ms	406	1.850%
Element_matches	1.570 ms	605	1.209%
Event_timeStamp_Getter	1.540 ms	500	1.186%
Element_getElementsByTagName	1.316 ms	503	1.013%

Blink RCS & Speedometer



Parsing	1158.565 ms	1101938	82.346%
ConstructTree	591.804 ms	394473	42.063%
DocumentFragmentParseHTML	167.762 ms	609	11.924%
ParserAppendChild	139.837 ms	310552	9.939%
HTMLTokenizerNextToken	137.166 ms	395695	9.749%
ReplaceChildrenWithFragment	121.996 ms	609	8.671%


blink
ContainerNode
RemoveChild
0.79s (0.86%)
of 37.25s (40.73%)
blink
HTMLFrameOwnerElement
PluginDisposeSuspendScope
~PluginDisposeSuspendScope
0.45s (0.49%)
of 0.99s (1.08%)
blink
ContainerNode
WillRemoveChild
0.51s (0.56%)
of 10.86s (11.88%)
blink
ContainerNode
RemoveBetween
0.74s (0.81%)
of 2.05s (2.24%)
blink
ContainerNode
NotifyNodeRemoved
0.74s (0.81%)
of 9.24s (10.10%)
blink
HTMLElement
AdjustDirectionalityIfNeededAfterC
0.46s (0.5%)
of 0.66s (0.72%)
blink
Node
NotifyMutationObserversNodeWillDetach
0.25s (0.27%)
of 0.33s (0.36%)
blink
DispatchChildRemovalEvents
0.41s (0.45%)
of 2.12s (2.32%)
blink
PluginsPendingDispose
0.34s (0.37%)
blink
TreeScope
AdoptIfNeeded
0.12s (0.13%)
of 0.45s (0.49%)
blink
Document
ExistingAXObjectCache
0.18s (0.2%)
of 0.50s (0.55%)
blink
Node
RemovedFrom
0.23s (0.25%)
of 0.69s (0.75%)
blink
probe
willRemoveDOMNode
1.05s (1.15%)
of 1.20s (1.31%)
blink
Document
GetProbeSink
0.38s (0.42%)
of 0.45s (0.49%)
blink
Element
CheckForEmptyStyleChange
0.45s (0.49%)
of 0.82s (0.9%)
blink
FrameSelection
NodeWillBeRemoved
0.24s (0.26%)
of 0.33s (0.36%)
blink
SelectionEditor
NodeWillBeRemoved
0.54s (0.59%)
of 0.58s (0.63%)
blink
Node
InActiveDocument
0.17s (0.19%)
of 1.31s (1.43%)
MutableComputedStyle
0.18s (0.2%)
of 0.42s (0.46%)
0.15s
0.10s
0.11s
0.17s
0.08s
0.84s
10.76s
1.94s
9.07s
0.58s
0.05s
0.10s
0.09s
0.12s
0.15s
6.16s
0.31s
2.08s
0.22s
0.30s
0.36s
0.50s
0.54s
0.70s
(inline)
5.77s
0.10s
0.13s
0.09s
0.14s
0.73s
1.07s
0.45s
0.59s
0.17s
(inline)
0.46s
0.10s
0.10s
0.12s
(inline)
0.06s
(inline)
0.06s
(inline)
0.06s
(inline)
0.08s
(inline)
0.23s
0.48s
0.26s
0.05s
(inline)
0.25s
(inline)
0.07s
(inline)
0.13s
(inline)
0.29s
(inline)
0.05s
(inline)
1.52s
(inline)
0.20s
(inline)
0.05s
(inline)
0.06s
(inline)
0.07s
0.80s
0.19s
0.15s
0.11s

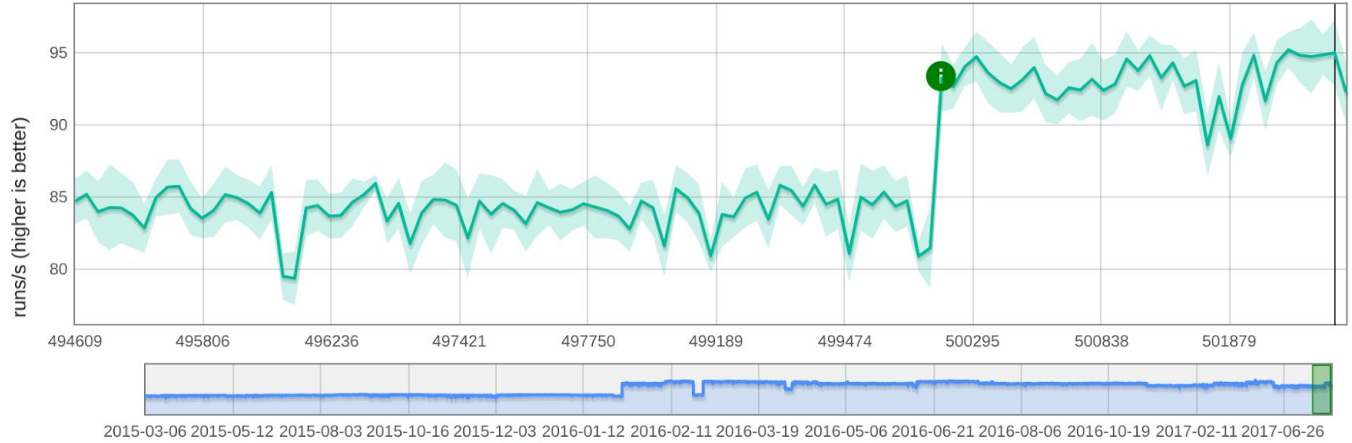
Send feedback

Test suite: blink_perf.events | Bot: android-one | Subtest: EventsDispatching | Subtest | ADD

Some charts failed to load. Either they don't exist, or they are empty, or they are accessible only for signed-in google.com accounts.

ChromiumPerf/android-one/blink_perf.events / EventsDispatching

CLOSE



Click and drag graph to measure or zoom.
Traces: select all | deselect all | core only

- [x] EventsDispatching
3 more

ChromiumPerf/linux-release/blink_perf.events / EventsDispatching

CLOSE

Click and drag graph to measure or zoom.
Traces: select all | deselect all | core only

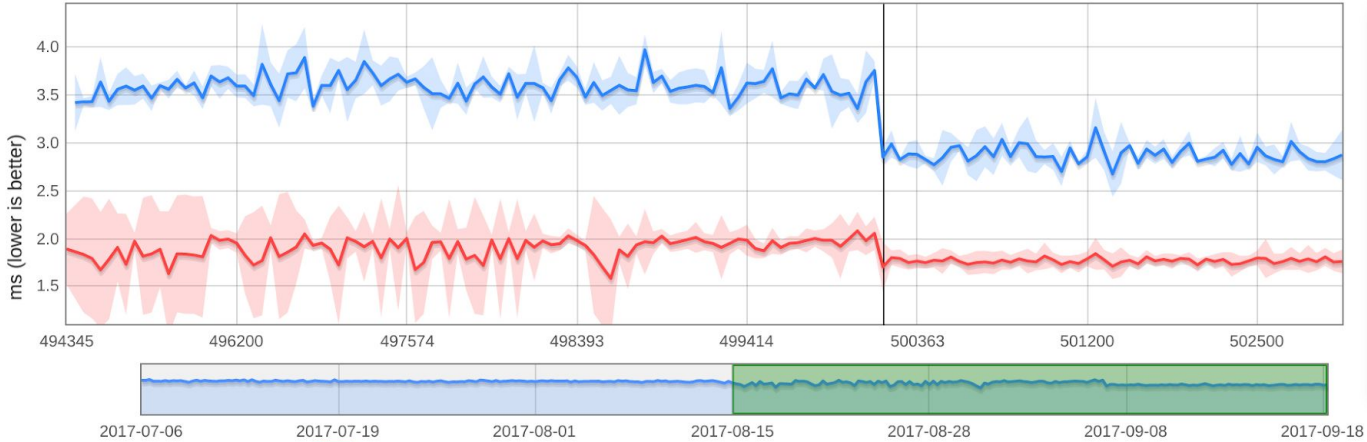
- [x] EventsDispatching
- [ ] ref
2 more

Test suite Bot ADD

ChromiumPerf/chromium-rel-mac11/v8.runtime_stats.top_25 / Blink_EventTargetDispatchEvent:duration_avg

CLOSE

v8.runtime_stats.top_25: Runtime Stats benchmark for a 25 top V8 web pages. Designed to represent a mix between top websites and a set of pages that have unique V8 characteristics.



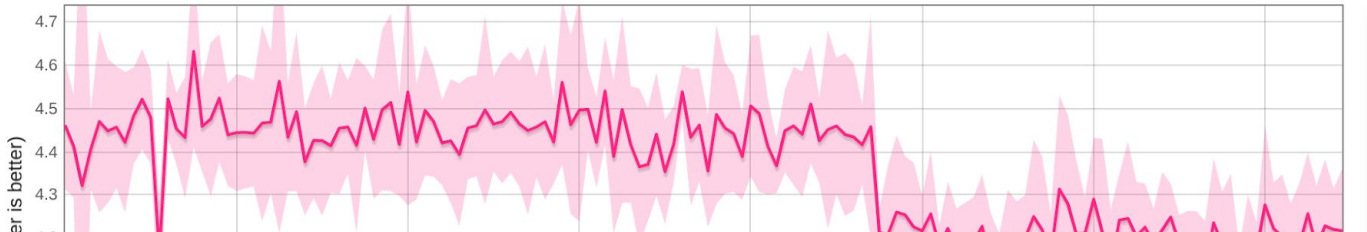
Click and drag graph to measure or zoom.
Traces: select all | deselect all | core only

- Blink_EventTargetDispatchEvent:duration_avg
- http___edition.cnn.com

6 more

ChromiumPerf/chromium-rel-mac11/v8.runtime_stats.top_25 / Blink_HTMLElementClick:duration_avg

CLOSE



Click and drag graph to measure or zoom.
Traces: select all | deselect all | core only

- Blink_HTMLElementClick:duration_avg

5 more

Next steps

- Use measurement and profiling results to **improve synchronous performance** (notably script, bindings and DOM performance)
- Compare in more detail to other implementations to **identify areas where better performance/score is known to be possible**
- Break down contributions of both **synchronous** and **asynchronous** work to the Speedometer score
- Monitor for **regressions**
- Keep an eye out for **Speedometer 2.0**

Thank you

ありがとうございました

platform-architecture-dev
@chromium.org

chromium