Firefox Performance Meeting - Google Docs **(Mozilla Public, Non-confidential)**

Meeting details

Meeting time, ZoomID: 452-1433-742. 30 minutes

Meeting purpose

Stand-ups amongst front-end engineers who are working on performance improvement related efforts. Opportunity to hear “the news” of what people are working on, and make group decisions. It’s also where we can group triage things that we can’t triage individually.

Note: For the most part, deep technical discussion on performance problems should be taken outside of the meeting, or left until the end.

Estimates

- Format: [fxperfsize:<SIZE>]
- <SIZE> = XS, S, M, L, XL
- XS: 1-2 days
- S: <= 1 week
- M: 1-2 weeks
- L: 1 month
- XL: > 1 month
- *JIra Story Points: using Fibonacci format for story points: XS- 3 points, S- 5 points, M - 8 points, L - 13 points, XL- 21 points*

Meeting structure template

Sep 23, 2025 | 📅 fx-perf quarterly meetup

Attendees: Florian Quèze Gijs Kruitbosch Mike Conley Alex Thayer

Notes

- [gijs] calixte is minifying PDFjs better, and seeing some perf improvements. PDFjs uses a worker script and so this may have outsized impact (might not be using xulcache? Unclear.)
    - [gijs] would we consider this for more JS? Calixte and others primarily looking at mobile next.

Action items

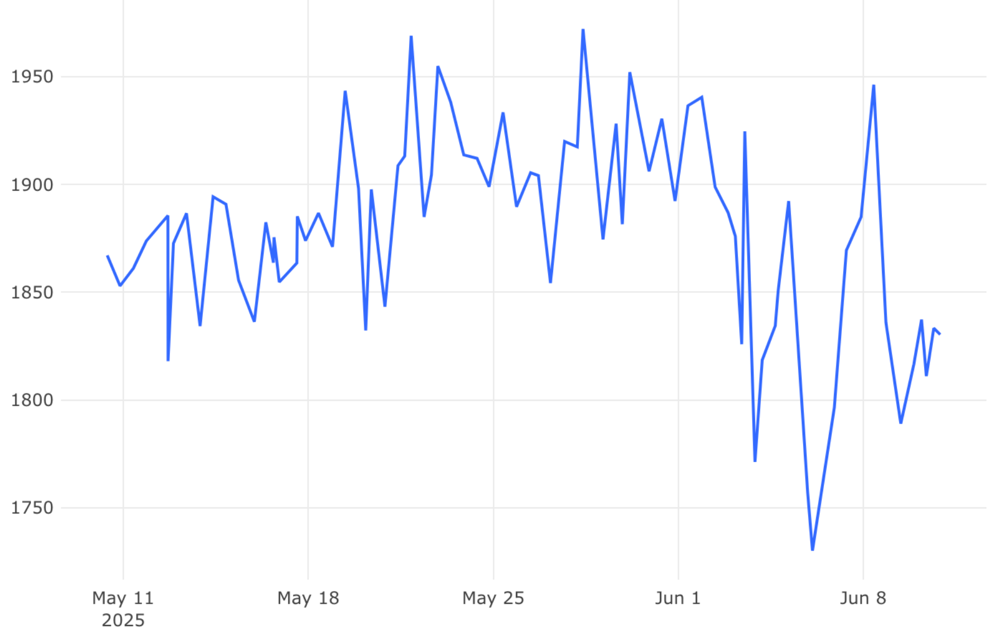
- □

Jun 11, 2025 | 📅 fx-perf quarterly meetup

Attendees: Florian Quèze Gijs Kruitbosch Mike Conley Alex Thayer

Notes

- [florian] removed the 300ms wait after a tab switch for browser chrome mochitests
    - Used a pref to revert to the old behavior in about 340 tests (out of 7k bc tests)
    - Seems to have reduced the average bc job time by about 1 minutes, although the data is very noisy: https://sql.telemetry.mozilla.org/queries/108253#265962



    - Planning to send a post to dev-platform / firefox-dev about this once I'm done greening up the tree (I'm still working the mac aarch64 1500 platform that's tier2 but makes sheriffs star lots of oranges)
- [florian] There's a patch on autoland to remove the 1s wait when starting other kinds of mochitests
- [florian] MOZ_PROCESS_SAMPLING=1 (on try: `mach try fuzzy —env MOZ_PROCESS_SAMPLING=1` shows markers for processes in resource use profiles.
    - Example profile showing what takes time in 'Ba' (artifact builds) jobs: https://share.firefox.dev/4nflfwP
- [florian] Looking at the pain points and productivity loss when trying to land code: 📄 From code written to code shipped in Nightly

- [florian] Currently trying to make getting profiles of xpcshell test failures on try easy. Would ideally like all tests to run with the profiler enabled by default, and upload profiles automatically on failure.
- [mconley] Re-looking at: https://phabricator.services.mozilla.com/D222757
- [gijs] Writing out session store data more efficiently
  - Also slack thread.

Action items

□

Mar 11, 2025 | fx-perf quarterly meetup

Attendees: Florian Quèze Gijs Kruitbosch Mike Conley Alex Thayer

Notes

- browser_startup.js intermittent test failures
  - This seems fairly brittle, and we only noticed something was going wrong because of an increase in intermittent failures. robwu suggests this may have been exacerbated by the removal of the screenshots system addon
- Any suggestions on 1945048 - severe browser lags and high CPU and RAM usage ?
- Alex demo'ing JS execution tracing profiler feature
- [florian] (read only) another cool new profiler thing: we can now profile the size of a binary, which is very useful to understand installer size regression bugs: https://github.com/jrmuizel/binary-size-profiler
  - Example: https://bugzilla.mozilla.org/show_bug.cgi?id=1944663#c3

Action items

☑ ~~Mike Conley to ask jmaher about exempting browser/base/content/test/performance from the "single tracking bug for intermittent failures" strategy, because we have clear examples of this strategy causing us to not notice problems. This should also be exempted from blanket-disabling of the test.~~

Jan 7, 2025 | 📅 fx-perf quarterly meetup

Attendees: Florian Quèze Gijs Kruitbosch Mike Conley ~~Alex Thayer~~

Notes

- [florian] BHR for content processes: https://fqueze.github.io/hang-stats/child/
  - Shocked by how much cycle collection is in there, filed https://bugzilla.mozilla.org/show_bug.cgi?id=1938084
  - YouTube bug: https://bugzilla.mozilla.org/show_bug.cgi?id=1935456
- [florian] JIRA discussion about BHR data retention
  - The proposal is to switch to 30 days.
  - Means we would give up on re-running failed jobs.
  - And also give up on queries like https://sql.telemetry.mozilla.org/queries/102988/source#253834
- [florian] What are build times like for you?
- [florian] Are we glean yet (70%)
- [florian] binary size investigation(?)

Action items

☐

Sep 20, 2024 | 📅 fx-perf quarterly meetup

- [mconley]
  - Bug 1918208 - 15 minute automatic session restore on startup after recent vertical tabs change
- [florian]
  - New profiler markers for data recorded in legacy telemetry Histograms, Scalars and Events. + markers for errors when trying to record histograms and scalars. + markers for ping submissions.
  - WIP profiler markers for data recorded through Glean.
  - Removed expired telemetry probes
  - Glean migration
- [Gijs] idle startup tasks to enable event recording

Jun 11, 2024 | 📅 fx-perf quarterly meetup

- [mconley]
    - Second jump list fix - making favicon caching work off-main-thread
    - Then removing the legacy backend.
- [florian]
    - Any new front-end engineer who might be interested in looking every once in a while at BHR data?
- [alexical]
    - Demo
- Action items:
    - mconley to reach out to Marco Castelluccio about potentially having some kind of ML model thing ingest BHR data and file new bugs for stacks that have no bugs associated with them and make sure they're tagged for Perf triage.
    -

Mar 19, 2024 | 📅 fx-perf quarterly meetup

- [mconley]
    - Windows Jump List thing goes out today!
    - Fix to improve startup time when doing paveover installs with the stub installer is going out in 125. This is the 30 second white screen that users with slower networks might experience.
    - fx-perf clearout is kinda at a standstill. Maybe this quarter?
- [florian] procmon
    - florian is working on making it possible to ingest markers from procmon and display them alongside captured profiles. This is in early stages, and florian is looking for
- [Gijs] Make it harder to introduce new mainthread I/O

Dec 12, 2023 | 📅 fx-perf quarterly meetup

Attendees: fqueze@mozilla.com gkruitbosch@mozilla.com mconley@mozilla.com dothayer@mozilla.com

Notes

- [florian] Startup

- Order file
    - Serge would like to remove them and generate them automatically during PGO builds, but has failed to notice any difference with/without order file on his fast machine.
    - [gijs] originally added in https://bugzilla.mozilla.org/show_bug.cgi?id=1542746
    - > Switch clang-based PGO builds from front-end based instrumentation to IR-level instrumentation.
    - [gijs/mconley] Are we still using the right optimization? Bug cites rust optimization concerns; are we no longer optimizing rust with serge's changes? Or did rust grow support for the other kind of optimization?
    - [gijs] use mach try perf?
    - [florian] expect no difference there due to cpu/disk characteristics
    - [mconley] we have startuprecorder jobs "fxrecord" that run on slow reference devices, should be able to trigger those on try and compare
    - [gijs] may also be worth comparing between the old file, new file _and no file at all_
    - [mconley] either way making it part of automation rather than having a static file seems beneficial.
- Startup flicker
    - Discussion today in #desktop-integrations (Slack) / #gfx (matrix)
    - [Slack thread]
    - [florian] tldr: there's some weird flicker on Windows, and there are some maintenance needs for the skeleton UI
    - [mconley] when do we show the skeleton UI?
    - [florian] we show unconditionally, issue is we should update it when the main browser UI layout changes
        -
    - [florian] other thing is the flicker on Windows that we don't understand.
        - [mconley] only the content area flickers
        - [gijs] there are some cases where we use transparency for tabs in the browser; I wonder if we're drawing that?
        - [mconley] tab background flickers too; maybe bg flickers but not foreground bits?
    - [mconley] Pascal claims we show the bad skeleton UI for a longer time than if we disable that the skeleton UI.
        - [florian] think this is user-perception issue
        - [mconley] maybe check with a screen capture?
    - [mconley] let's file something and tag Alex Thayer 🙂
        - [mconley] And needinfo Pascal to ask for a recording of the before/after to understand if there's something wonky about the skeleton UI feeling slow
- Window opening flicker

        - https://bugzilla.mozilla.org/show_bug.cgi?id=1792098
        - [florian] want to avoid disabling the test
        - [gijs] let's ping Emilio about the Linux failure?
        - [mconley] finding what's moving the mouse on macOS is probs interesting… I guess we don't have a good way
        - [florian] kind of want a profile for both cases
    - js modules loaded during startup.
        - 8 months without fixes is long in https://bugzilla.mozilla.org/show_bug.cgi?id=1828735
    - 30s delay
        - 1749345 - White screen for up to 30 sec after updating Firefox using the full installer, due to FirstStartup component
- BHR
    - [mconley] I know that jrmuizel is interested in more ways to measure browser responsiveness. I reminded him about BHR.
- [mconley]
    - WindowsJumpList stuff
        - Added a new backend that commits jump lists off of the main thread
        - Enabled by default, held to Nightly
            - Once this rides the trains, I plan on doing a controlled rollout to ensure stability stays constant.
- [mconley] fxperf clearout
    - Whoops, I guess we mostly forgot to do this. There are 99 bugs left. Maybe we can get it done before the end of the year? Bug lists remain the same.

Action items

☑ ~~mconley to file something about the skeleton UI feeling slower (Done)~~

Sep 21, 2023

- [mconley] fx-perf cleanout
    - Down to just 103 bugs in [fxperf:p3]! Hooray! Okay, going to redistribute the rest, and hopefully we can clear these all out before our next meeting and be done with it.
- [mconley] On the side, I've been trying to move Windows jump lists fully off of the main thread in this bug. I'm writing GTests now. Slow going, but making progress.

Jul 28, 2023

Notes:

- [mconley] about:home startup cache is finally out on release at 100%. Phew.
- [mconley] fx-perf cleanout
- [florian] Eco Mode
    - This idea keeps coming back and leadership doesn't act on it. It's difficult to convince anybody in leadership that we should work on this.
    - We're trying to measure how much power we can save and trying to measure user interest in the feature.
    - We're trying to get numbers an upper bound of power savings on Nightly (100%). There's no visible user experience change here (unless we were too aggressive - there was a small handful?).
    - The ideas we tested:
        - Reducing framework (30fps)
        - Disabling video autoplay
        - Throttling timers in background tabs
    - Q: Has anybody talked about how this might be important for people who run software that are intensive, like video editing? Like, a way of making Firefox less hungry if it's not the primary piece of software running.
        - A: Not really… we don't think throttling the background timers is ultimately going to have the biggest impact. Telemetry suggests 8% of activity being from background tabs.
    - Q: Tab unloading? Did we try playing with that?
        - A claim on macOS is that we never get a low memory notification, so this never really happens. So Firefox will use 10's of GB of memory and Firefox is paging really hard. Recent discussion suggests that TabUnloader isn't doing a very good job.
        - We didn't consider this - this causes data loss, potentially. We, instead, could "suspend" the tab (this is what Android does). This doesn't do much for memory, but would help with memory.
- [florian] Bug 1838497 - Add a profiler feature to record the clock frequency of every core while sampling
    - Example profile: https://main--perf-html.netlify.app/public/z1mz2rkpwqn0hwm1pp40zn5nns4h07d0q4errk8/calltree/?globalTrackOrder=80w7&hiddenGlobalTracks=1w5&hiddenLocalTracksByPid=13000-0w47wb~4708-01~17016-01~17008-01~16828-01~17028-01~9200-01356~3088-2&thread=g&v=10

Mar 14, 2023

Notes

- [mconley] fx-perf cleanout
- [mconley] about:home startup cache
    - Experiment results and analysis
    - Huzzah, finally, this thing is going to roll out! Justine Oliver is a PM who's helping this go out with a hold-back experiment.
- [florian] can we disable about:home updates while the user is away? https://phabricator.services.mozilla.com/D172172
    - Scott Downe
- Unfiltered link: https://share.firefox.dev/3mC1qEY (filtered to the update timers: https://share.firefox.dev/41Wu8Re)

Dec 13, 2022 |

Attendees:

Notes

- [mconley] about:home startup cache
    - about:home startup cache - Nimbus Experiment Brief
    - Experiment 3 underway on release channel
- [mconley] Bug 1529276 - resource:///modules/WindowsJumpLists.jsm still does main thread I/O
    - Making slow but steady progress on this
- [florian] Could we make Joy of Profiling open session meetings more useful? (ie. convince more people to join)
    - [mconley] What's attendance like?
    - There is this document for DevTools. Maybe this should be a general Testing document (like this one), and we augment that with information on how to use the Profiler to investigate intermittents.
- [florian] ownership of front-end performance tests
    - browser/base/content/test/performance
    - Now owned by Florian as triage owner as part of Toolkit :: Performance Monitoring
    - Triage rotation
- [florian] I would like to find an automated way to verify that we don't have left over repeating timers. Thinking about something similar to what I did with vsync.
    - Crashy first attempt on try
- [florian] Status update points (can be readonly)
    - Power profiling supported on Linux, on Intel Macs, and on AMD CPUs (Windows 11 / Linux).

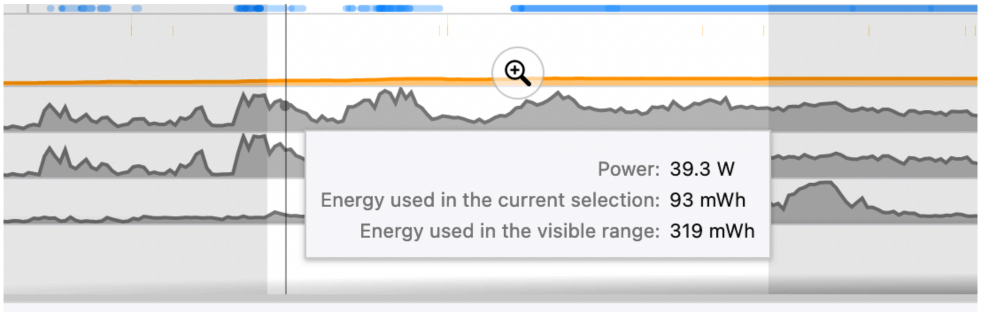
    - Efficiency mode for background content processes on Windows 11.
    - Timer work is still happening despite Gerald leaving. Justin Link took over.
    - We have even more interesting Glean probes to understand our power use.
        - CPU names (https://sql.telemetry.mozilla.org/queries/88505)
        - (nika) IPC message count (https://sql.telemetry.mozilla.org/queries/89064#220647)
- [dthayer] ochameau has a patch stack which moves devtools out of the parent process, but it's tricky because we have so many tests that depend on it being in the parent process. 🎉🎉🎉
    -
- [dthayer] Been working on Speedometer 3, both in terms of working on new Speedometer benchmarks as well as JS/elsewhere optimizations for the benchmarks we have. It would be nice if we could get an extra win for our front-end by optimizing for Speedometer. If we have any ideas for a kind of benchmark that might represent a performance-problematic piece of our frontend, that could be valuable.
    - [mconley] to rephrase: can you design a benchmark that's webby and works in SPD3 that we are confident translates to improvements in browser UI JS?
    - [dthayer] yes - e.g. replacing JSON.parse/stringify with structuredclone - this would help the spidermonkey team prioritize working on it and then that'd improve things for desktop perf too.
    -
    - [mconley] large dynamically generated tables (e.g. about:config)
        - [dthayer] minecraft clone using tables
    - [gijs] can we make sure the events driving the tests are "real"? See e.g. this speedometer regression that I tripped by changing handling of untrusted click events (which got tripped by speedometer)
        - [dthayer] yeah… but we want to make sure it can run on the web, so we aren't able to do everything we like. The "honour system" here is sort of the best we got
- Actions
    - Mike Conley to invite Florian to the Front-end Sync up meeting to demonstrate how to use the profiler to investigate intermittents.
    - Gijs Kruitbosch to setup triage rotation for performance monitoring / static performance tests
    - Doug Thayer to raise to Patricia / Maire that there's an opportunity to improve performance of DevTools by putting them out of the parent process.

Sep 13, 2022 | 📅 fx-perf quartarly meetup

Attendees: Florian Quèze Mike Conley Doug Thayer

Notes

- Power profiling
  - Available on Windows 11 with Intel CPUs, and Apple Silicon. Working on making it available on Linux.
  - It got some attention when it shipped in Firefox 104, including a blog post that reversed engineered it and wonders if/when a similar feature will appear in other browsers.
  - In addition to showing the instantaneous power, the profiler tooltip for power tracks now shows the energy used over the selected time range.



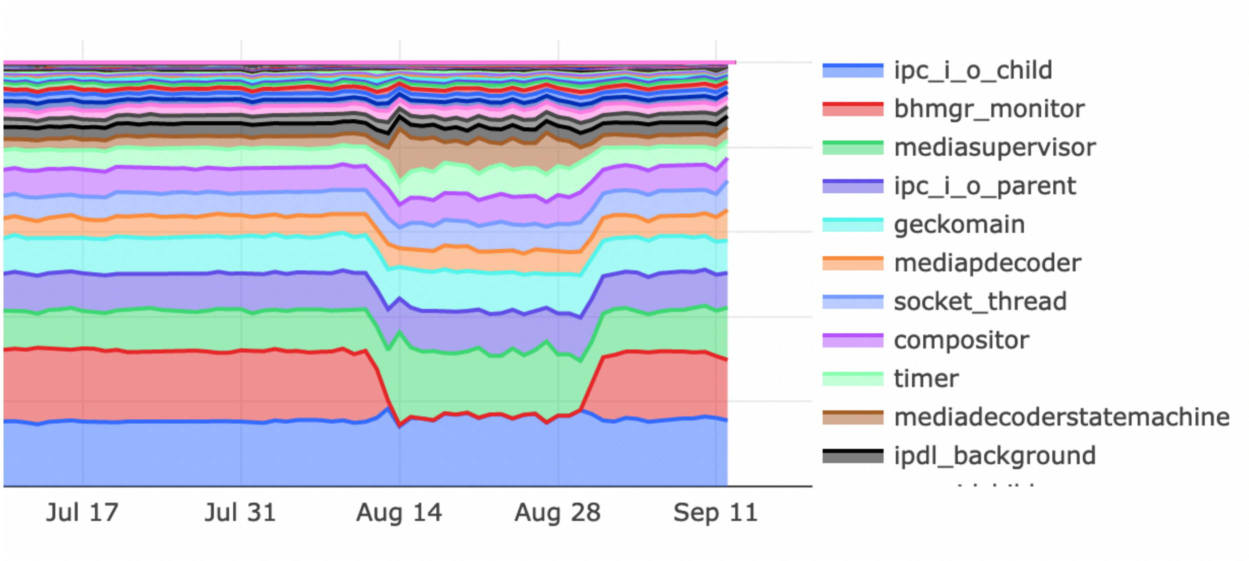
Power: 39.3 W

Energy used in the current selection: 93 mWh

Energy used in the visible range: 319 mWh

- Blinking cursor GPU power bug
- Power use wins
- Think about mobile: is Firefox being killed when in the background more often than competing browser?
- Telemetry dashboards for power:
  https://sql.telemetry.mozilla.org/dashboard/firefox-desktop-resource-use
  https://sql.telemetry.mozilla.org/dashboard/cpu-use-on-nightly
- Hoping to run some A/B testing related to power use during H2. Most likely will test the impact of reducing the frame rate, and the impact of lowering priorities for trackers.
  - Talk to James Stephens II

- BHR without so many wake-ups. Landed and got backed out after 3 weeks for a perma-fail (no kidding).



- about:home startup cache (yep, coming back to this)
  - about:home startup cache - Nimbus Experiment Brief
- Bug 1529276 - resource:///modules/WindowsJumpLists.jsm still does main thread I/O
- Gerald leaving :-(. If you need any question answered related to how the profiler platform code works, ask today.

Action items

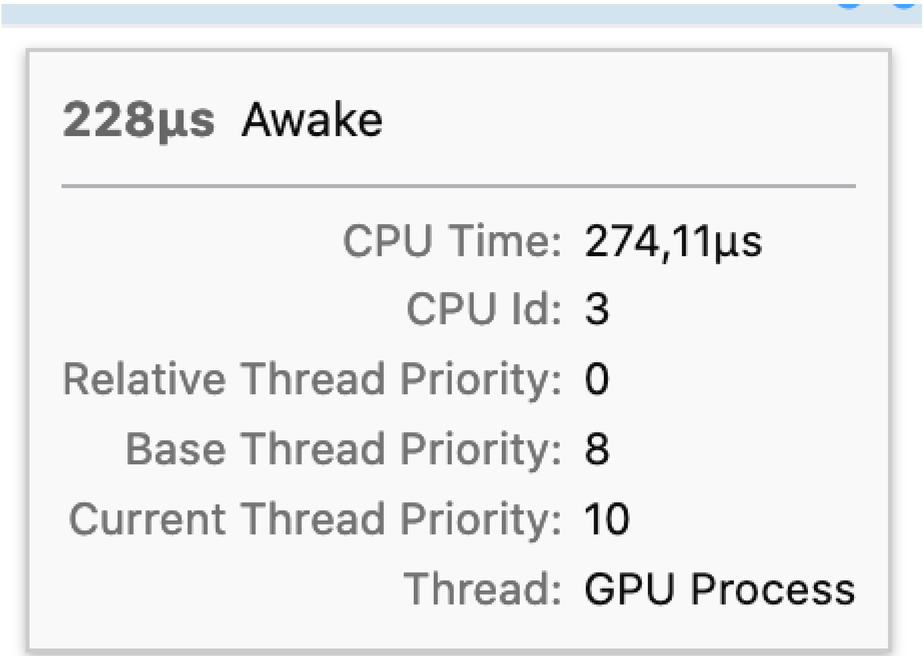
- ☐ Florian to find or write a doc about eco mode

March 8th, 2022

- We continue to be mothballed until further notice.
- New Bugzilla keywords: https://wiki.mozilla.org/Performance/Bugzilla
  - Also see https://wiki.mozilla.org/Performance/Triage
  - Process:
    - Add perf:frontend keyword to all bugs with [fxperf:*]
      - 94 bugs with [fxperf:p2]
      - 212 bugs with [fxperf:p3]
      - ~300 bugs total. Can we commit to having all of these migrated to the new keywords by the end of the month?
        - This will mainly be a task of making sure that they actually belong to perf:frontend, and adding any additional

keywords as necessary. Then setting a performance level on it.

    - Add perf:startup keyword to all startup bugs (this can co-exist with perf:frontend)
  - Performance Priority != Bug Priority
    - Example
- Bug 1742797 - composites happen at 60Hz until the browser window is closed after showing an arrowpanel
  - How do we get this over the line?
- Thread and Process priorities: https://share.firefox.dev/3JKBCNa



228µs Awake

CPU Time: 274,11µs
CPU Id: 3
Relative Thread Priority: 0
Base Thread Priority: 8
Current Thread Priority: 10
Thread: GPU Process

  -
- Dll loads: https://share.firefox.dev/3w5Gd9h
- Power use work:
  - Sockets on Mac
  - Timers API
  - Awake markers, and hoping to add telemetry for wake ups.

January 13th, 2022

- We continue to be mothballed until further notice.
  - Core Actives
  - We hear the phrase “performance is tablestakes”, but we don’t appear to be investing in it right now. How do we reinvigorate that.
  - “Quality” might be a theme for MR2, says florian.
- What kind of ways could we compete on Chrome with?

        - dthayer has some ideas here. There was a meeting where everyone came up with an item of things that annoyed them about Firefox. dthayer wants to also document the things that annoy him when using Chrome.
- Monthly presence from Desktop requested at Biweekly Performance Strategy Meeting
    - Any volunteers? If not, will create some kind of rotation.
    - dthayer is already there, mconley will attend once a month
- fx-perf triage
    - [mconley] Have actually done this, but usually the list is empty. Will poke the bi-weekly meeting about this. Since we're not focused primarily on startup these days, my advice is to try to identify power-related things and get Florian's attention.
- Power usage updates from florian
    - Telemetry for CPU and GPU time used, https://sql.telemetry.mozilla.org/dashboard/carbon-footprint-desktop-indicators
        - About 2 billion minutes of CPU used per day. That's about 3800years of CPU time spent everyday.
    - Thinking a lot lately about excessive CPU wakeups
        - Ensuring vsync doesn't remain enabled would help.
        - Found other bugs causing lots of wakeups
    - Trying to identify high value self-contained projects that could help with power
        - Making Timers and IPCs collaborate to reduce wakeups
        - Reducing CPU use of the Socket Thread on Mac
        - Efficiency cores.

December 14th, 2021 (Read-only - meeting deferred until 2022)

- We continue to be mothballed until further notice.
    - [mconley] Have mostly forgotten about sending out triage emails. Apologies. :/
    - We lost emalysz, but have (re)gained mcheang. Might be a good opportunity to rope her into fx-perf stuff if and when we get revived.
- Bas reached out and wants to chat with a few of us before the year ends to coordinate for next year.
- Power usage updates from Florian?
    - Collecting Telemetry data to understand our power use
        - Collecting total CPU and GPU time used:
            - CPU time recorded in:
                - parent and content process (reached release)
                - GPU process (landed on Nightly on Dec 1st, already on beta)
                - RDD process (landed on Nightly on Dec 10)
                - Socket process (in review)
                - GMP process (in review)
                - VR not recorded

                - REMOTESANDBOXBROKER not recorded (only used on Windows ARM)
            - GPU time recorded on:
                - parent process (already on beta)
                - GPU process (already on beta)
                - RDD process (still on nightly only)
        - Dashboards:
            - CPU time and bandwidth used: https://sql.telemetry.mozilla.org/dashboard/carbon-footprint-desktop-indicator
                - 110min of CPU time per user per day on Nightly, 56min on release. The release channel uses a total of over 2 billion minutes of CPU time per day.
                - 12 million hours of videos played per day.
                - 25 million Gb of data transferred per day.
            - GPU time per user: https://sql.telemetry.mozilla.org/queries/83200#206203
                - 60 min of GPU time per day for Nightly Windows users.
        - Planning to add more detailed probes to have a breakdown per process type, for foreground/background content processes. Maybe per runnable (we only have runnable names on Nightly).
            - Would like to understand how much we should focus on reducing power use while Firefox is completely unused vs power use of background tabs, vs power use of active use.
    - Chasing unintentional 60Hz animations.
        - Would like to have all browser chrome mochitests ensure vsync is disabled at the end.

September 14th, 2021

Operations and updates (please add your news!)

- Thanks for coming together! We continue to be mothballed until further notice.
    - Great seeing our fx-perf skills/expertise being put to good use in Pro Client
- Fission is beginning its rollout. With that, Gijs has filed some bugs to make process flipping after session restore less likely.
- We're seeing a very slow but steady decline in the number of tab switch spinners users on Nightly are experiencing. Possible explanations are:
    - WebRender rollout, and the removal of old Layers paths
    - Fission rollout
    - Hardware churn
    - Old builds providing tab switch spinner data, slowly polluting the data set

- Out-of-process filepicker has stalled out because aklotz has left
    - Patch set is still WIP and was never reviewed. Here’s a snapshot on try.
    - This still blocks this bug, which is in the top 3 on hang stats
- XULStore / LMDB main thread IO seems to have been fixed
- Open floor
    - [florian] Power use
        - [mconley] reduce refresh rates
        - Opportunities for better scheduling with task controller?
    - One-click profiling from about:processes
        -
    - [emalysz] backlog/triage
        - [fxperf:<p>:resource]
            - resource
            - startup
            - responsiveness

June 22, 2021

Operations

- Mothballing fx-perf during MR2
    - Triage
- Next desktop meeting speaker is: mconley for the foreseeable future!
- Arthur Edelstein has been tapped to be PM for Performance-related things

Engineering Updates

- bigiri
    -
- dthayer
    -
- emalysz
    - [read-only due to crisp meeting]
    - MR2 focus, so no real update
- florian
    - I removed the “Task Tracer” feature of the profiler. I never saw it work; it was a leftover from B2G performance efforts.
    - Imported Linux perf profiles will be a bit more useful, as the library names will be shown.
- Gijs
    -
- mconley

    -

Group Triage

-

Roundtable

-

June 15th, 2021

Operations

- And we’re back! Sorry about all of the cancellations!
- Next desktop meeting speaker is: bigiri

Engineering Updates

- bigiri
    - MR1.1
- dthayer
    - MR2
    - Working with James Teh on a fix for a problem with the skeleton UI and screenreaders (skeleton UI had to be backed out 😭)
- emalysz
    - Closed out a blocker that will enable us to override a backup file that already exists with IOUtils.write
    - Debugging xpcshell failures with ExtensionParent in order to remove OS.File from JSONFile.jsm
    - Barret is taking a look at a SessionWorker timeout and SessionFile regression
- florian
    - Doug added support for filtering by annotation on the BHR dashboard: https://fqueze.github.io/hang-stats/#row=0&annotations=UserInteracting%3Atrue%2C!ExternalCPUHigh%3Atrue shows hangs while the user was interacting and there was no high external CPU use.
- Gijs
    - (Away on PTO)
- mconley
    - Starting to dig into MR2 stuff. Along with some navigational model explorations, I’m also thinking about how TabUnloader could work with MR2. The idea being to

    - re-use TabUnloader to unload Views to reclaim memory while minimizing how annoying those unloads are.
    - Landed the about:home startup cache Nimbus switch with a test. Going to request uplift later today to see if I can run the experiment in at least Beta, but maybe Release.

Group Triage

-

Roundtable

-

June 1st, 2021

Operations

- Next desktop meeting speaker is:

Engineering Updates

- bigiri
    -
- dthayer
    -
- emalysz
    -
- florian
    -
- Gijs
    -
- mconley
    -

Group Triage

-

Roundtable

-

May 18th, 2021

Operations

- Next desktop meeting speaker is: Gijs
- mconley is away next week. Cancelling meeting for next week.

Engineering Updates

- bigiri
    - Proton
- dthayer
    - Have a patch up for recording js stacks in late write reports (for advancing fast shutdown)
    - Working on optimizations / perf analysis for using fathom to autofill credit card fields
    - MR2 poking around
- emalysz
    - Shutdown hangs from using IOUtils in PageThumbs
        - Throws an error
- florian
    - (long week-end last week - no update)
- Gijs
    -
- mconley
    - Getting roped into MR2
    - Putting together a guidelines document for fonts
        - [florian] FYI, there's ongoing work to load fonts off main thread during startup: https://bugzilla.mozilla.org/show_bug.cgi?id=1708768

Group Triage

-

Roundtable

-

May 11th, 2021

Operations

- Next desktop meeting speaker is: florian

Engineering Updates

- bigiri
  - Proton
- dthayer
  - PTO, reviews
  - Modified the BHR job to now output hang annotation information
  - Have a PR up for Florian’s tool to display this information in an easy-to-digest manner
  - Looked into an issue where apparently we only give 32-bit builds of beta to users on Windows?
- emalysz
  - [read-only]
  - Landed a patch to provide async support for promise workers
  - Removed OS.File from PageThumbs.jsm
  - 3 bugs remaining!
    - 2 R+ but failing tests, 1 was backed out for a perf regression
- florian
  - about:processes patches
    - Less wordy (removing plenty of redundant “Process” words)
    - Easier to find which threads are active without needing to expand individual processes
    - Thanks Doug for many reviews!
  - Profiler buffer memory is no longer counted in the profiler memory tracks, making it possible to see small memory use variations.
  - New label frames for XPIDL method/getter/setter calls (similar to what we already had for WebIDL)
  - H2 planning
- Gijs
  - MR1…
  - Thanks to Florian and the gfx team for help with triaging this perf bug and how to address it. Might be better on today’s nightly?
- mconley
  - Fixed Bug 1703494 - Remove sync layout flush for hamburger menu opening with proton - thanks emalysz for the quick review!

  - Talking with bwinton about packaging about:home startup cache enabled in MR1.1
  - Pro-tip for memory savings: nika is landing patches that make it possible to opt-out of JSActor creation on events that are only for cleanup.
  - In other news, mccr8 made it so that the process priority manager works for subframes / Fission!
  - PTO coming up towards the end of the month

Group Triage

-

Roundtable

- [florian] We (Bas, Markus and me) have been doing Joy of Profiling sessions on Zoom every Monday for a few weeks. Feel free to invite people who would like to get better at profiling, or who need help looking at profiles they have.

May 4th, 2021

Operations

- Next desktop meeting speaker is: florian

Engineering Updates

- bigiri
  - Still proton.
- dthayer
  - qf:p1 bug bash last week
    - Bug 1677509 - SendReceiveMouseInputEvent (sync IPC) shows up pretty high on the BHR list. It’s a bit mysterious, and we have a number of profiles showing long delays where the GPU is not responding to the events despite none of the relevant threads being busy.
      - Have a patch which aims to address one potential issue we noticed where we make a syscall for every IPC message in the queue.
      - (Thanks Florian for reprofiling several times on the 2018 ref hw)
    - Bug 1546847 - SessionStore sending its state to a worker can jank the browser for multiple seconds, according to BHR data.
  - [kmoir] summary of feedback on bug bash week
    - Good to work with different people, fresh eyes on bugs, watching people use profiler was helpful.
    - Long meetings were sometimes draining

- Good for smaller projects sometimes difficult for gecko where components require a lot of context.
- Difficult when you get to non-actionable bugs. Lacking context to take action on triage items. Could do triage separate from working on the bugs.
- Can we do this in a non no-meeting week
- Consider it like a triage rotation instead of involving everyone at once.
- Consider triaging bugs within a specific area and then inviting someone from that area to help. Example triage DOM related bugs, have someone from the DOM team to provide perspective. Reduces friction between component teams and performance if someone from the performance team is there. More likely to help if the performance team representative is there instead of just asking a component team to prioritize bugs.

- emalysz
  - Splitting out a patch to make promise workers async
  - Revisited removing OS.File from PageThumbs
  - Removing OS.File from JSONFile.jsm got backed out for a xpcshell failure
- florian
  - Profiler: showing category colors for individual frames in stacks shown in tooltips (screenshot shown last week) has been deployed.
  - Profiling mochitests that trigger customizationchange events (by adding/removing browser toolbar icons) now works.
  - Starting the profiler on Linux without the “stackwalk” feature will no longer block on initializing Lul (which can take multiple seconds), making automated tests capturing profiles (eg. our startup main thread I/O tests) much faster.
  - BHR data improvements:
    - AUTO_PROFILER_LABEL_DYNAMIC_NSCSTRING_NONSENSITIVE now correctly records a label frame in BHR, even when the profiler is not running.
      - This means these label frames are now visible: “ChromeUtils::Import”, “mozJSSubScriptLoader::DoLoadSubScriptWithOptions”, “nsThreadManager::SpinEventLoop”, “Category observer notification”, “Services_Resolve”, “Task”,
    - (chrome) JS function names are now included in BHR data.
  - BHR dashboard improvements:
    - automatic fallback to the data file of the previous day if the current one is empty
    - "previous day" and "most recent data" links at the top
    - the URL contains the day of the reports you looked at, so that if you share a link, people clicking the link will see the same data, even if they click the link on a different day.

      - This enables verifying the impact of recently landed patches on BHR data. Eg. Bas recently worked on reducing GetWorkspaceID hangs on Windows:
        - https://fqueze.github.io/hang-stats/#date=20210428&row=0&filter=nsWindow%3A%3AGetWorkspaceID -> 5256s of hang, 0.71% of total hang time. https://fqueze.github.io/hang-stats/#date=20210429&row=&filter=nsWindow%3A%3AGetWorkspaceID -> 905s of hang, 0.15% of total hang time.
    - For hangs with an associated bug, the whiteboard annotation is shown in the top right
    - when using the filter box, the filtered word is highlighted in the stack on the right side.
    - fetching the list of bugs and fetching the hang JSON file now happen in parallel
    - loading the page in a background tab works
- Gijs
  -
- mconley
  - Gearing down slightly from intense Proton work. Still working on it though, but starting to free up.
  - Aiming to get some time to land patches that would allow us to experiment with the about:home startup cache using Nimbus
    - k88hudson had consulted on this, and it looks like Nimbus works differently from Normandy, and would allow us to do this kind of experiment correctly.
    - And if all else fails and we can’t get it to work with Nimbus, then I think we should just let it ride.
  -

Group Triage

-

Roundtable

-

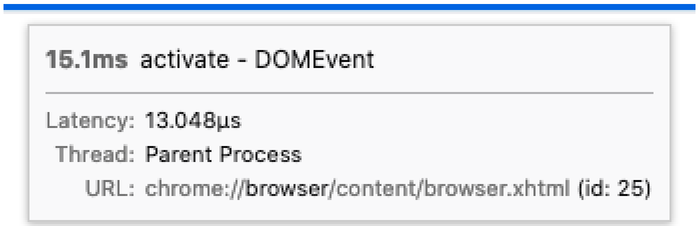
April 20, 2021

Operations

- Next desktop meeting speaker is: emalysz

Engineering Updates

- bigiri
    -
- dthayer
    - Fixed a bug where the skeleton UI wouldn’t show up when updating Firefox
    - Turned off skeleton UI for non-default UI density
    - Fixed a bug introduced by fixing a white flash in dark mode startup
    - Landed updates to skeleton UI to match Proton
    - Flipped pref for skeleton UI to ride the trains to release
    - PTO
- emalysz
    - Skeleton UI reviews
    - Proton
- florian
    - Fixed bug 1704868 - The notificationbar "stack" lazy getter is called when closing a tab
        - When I profiled closing a large number of tabs, this represented a bit more than 2% of the unresponsive time in the parent process.
        - Was a regression from proton-infobars work.
    - Added Worker.postMessage profiler markers, and label frames (that will be visible in BHR). Example profile: https://share.firefox.dev/3dzemnZ
        - I’m hoping to use this to understand how much of the JS Structured Cloning hangs we see on BHR are due to the SessionStore worker.
    - Added “Image load” and “Image paint” new markers.
    - I have patches up for review to update the about:processes table in a much more efficient way.
    - Profiler update deployed today, new things:
        - Profiling tests:
            - Put markers of the Test category first in the Marker Chart, as they are very relevant when they exist
            - Show the inner windows id in marker tooltips when there are multiple pages with the same url

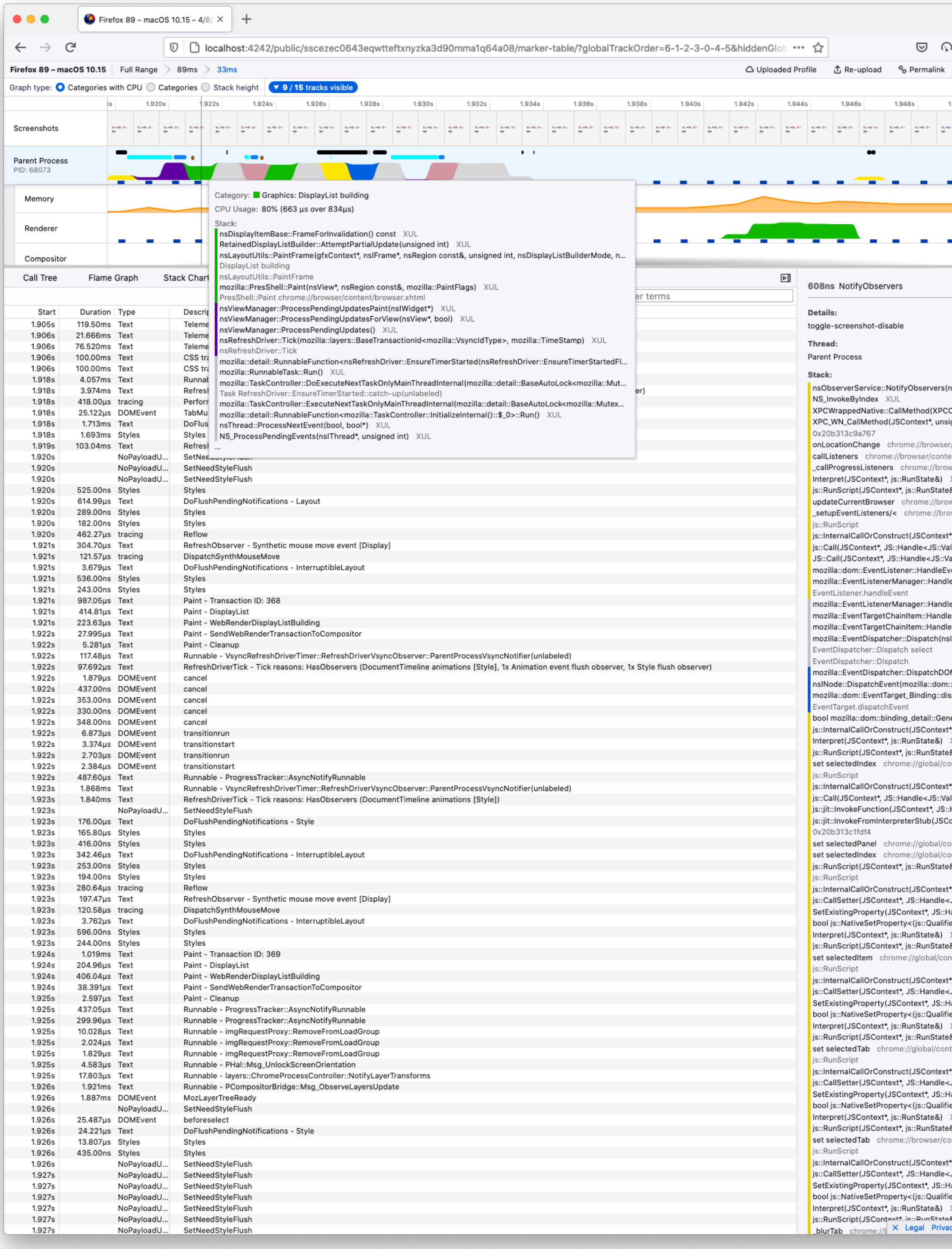


15.1ms activate - DOMEvent

Latency: 13.048µs
Thread: Parent Process
URL: chrome://browser/content/browser.xhtml (id: 25)

            - Treat profiles from a 127.0.0.1 url as local profiles

    - Accept gzipped data for profiles loaded from urls and local files with unknown extensions
  - Info panel:
    - Use an 'info' icon instead of the cloud icons for the profile info button
    - Add some vertical space between the 'Features' and 'Threads Filter' lists in the meta data panel
  - Talos profiles: Automatically expand all children in the zip file viewer
- Work in progress (up for review) : showing category colors for individual frames in stacks shown in tooltips:



Category: 🟩 Graphics: DisplayList building

CPU Usage: 80% (663 µs over 834µs)

Stack:

```
nsDisplayItemBase::FrameForInvalidation() const   XUL
RetainedDisplayListBuilder::AttemptPartialUpdate(unsigned int)   XUL
nsLayoutUtils::PaintFrame(gfxContext*, nsIFrame*, nsRegion const&, unsigned int, nsDisplayListBuilderMode, n...
DisplayList building
nsLayoutUtils::PaintFrame
mozilla::PresShell::Paint(nsView*, nsRegion const&, mozilla::PaintFlags)   XUL
PresShell::Paint chrome://browser/content/browser.xhtml
nsViewManager::ProcessPendingUpdatesPaint(nsIWidget*)   XUL
nsViewManager::ProcessPendingUpdatesForView(nsView*, bool)   XUL
nsViewManager::ProcessPendingUpdates()   XUL
nsRefreshDriver::Tick(mozilla::layers::BaseTransactionId<mozilla::VsyncIdType>, mozilla::TimeStamp)   XUL
nsRefreshDriver::Tick
mozilla::detail::RunnableFunction<nsRefreshDriver::EnsureTimerStarted(nsRefreshDriver::EnsureTimerStartedFl...
mozilla::RunnableTask::Run()   XUL
mozilla::TaskController::DoExecuteNextTaskOnlyMainThreadInternal(mozilla::detail::BaseAutoLock<mozilla::Mut...
Task RefreshDriver::EnsureTimerStarted::catch-up(unlabeled)
mozilla::TaskController::ExecuteNextTaskOnlyMainThreadInternal(mozilla::detail::BaseAutoLock<mozilla::Mutex...
mozilla::detail::RunnableFunction<mozilla::TaskController::InitializeInternal()::$_0>::Run()   XUL
nsThread::ProcessNextEvent(bool, bool*)   XUL
NS_ProcessPendingEvents(nsIThread*, unsigned int)   XUL
...
```

Start	Duration	Type	Descrip
1.905s	119.50ms	Text	Teleme
1.906s	21.666ms	Text	Teleme
1.906s	76.520ms	Text	Teleme
1.906s	100.00ms	Text	CSS tra
1.906s	100.00ms	Text	CSS tra
1.918s	4.057ms	Text	Runnab
1.918s	3.974ms	Text	Refres er)
1.918s	418.00µs	tracing	Perforr
1.918s	25.122µs	DOMEvent	TabMu
1.918s	1.713ms	Text	DoFlus
1.918s	1.693ms	Styles	Styles
1.919s	103.04ms	Text	Refres
1.920s		NoPayloadU...	SetNe
1.920s		NoPayloadU...	SetNeedStyleFlush
1.920s		NoPayloadU...	SetNeedStyleFlush
1.920s	525.00ns	Styles	Styles
1.920s	614.99µs	Text	DoFlushPendingNotifications - Layout
1.920s	289.00ns	Styles	Styles
1.920s	182.00ns	Styles	Styles
1.920s	462.27µs	tracing	Reflow
1.921s	304.70µs	Text	RefreshObserver - Synthetic mouse move event [Display]
1.921s	121.57µs	tracing	DispatchSynthMouseMove
1.921s	3.679µs	Text	DoFlushPendingNotifications - InterruptibleLayout
1.921s	536.00ns	Styles	Styles
1.921s	243.00ns	Styles	Styles
1.921s	987.05µs	Text	Paint - Transaction ID: 368
1.921s	414.81µs	Text	Paint - DisplayList
1.921s	223.63µs	Text	Paint - WebRenderDisplayListBuilding
1.922s	27.995µs	Text	Paint - SendWebRenderTransactionToCompositor
1.922s	5.281µs	Text	Paint - Cleanup
1.922s	117.48µs	Text	Runnable - VsyncRefreshDriverTimer::RefreshDriverVsyncObserver::ParentProcessVsyncNotifier(unlabeled)
1.922s	97.692µs	Text	RefreshDriverTick - Tick reasons: HasObservers (DocumentTimeline animations [Style], 1x Animation event flush observer, 1x Style flush observer)
1.922s	1.879µs	DOMEvent	cancel
1.922s	437.00ns	DOMEvent	cancel
1.922s	353.00ns	DOMEvent	cancel
1.922s	330.00ns	DOMEvent	cancel
1.922s	348.00ns	DOMEvent	cancel
1.922s	6.873µs	DOMEvent	transitionrun
1.922s	3.374µs	DOMEvent	transitionstart
1.922s	2.703µs	DOMEvent	transitionrun
1.922s	2.384µs	DOMEvent	transitionstart
1.922s	487.60µs	Text	Runnable - ProgressTracker::AsyncNotifyRunnable
1.923s	1.868ms	Text	Runnable - VsyncRefreshDriverTimer::RefreshDriverVsyncObserver::ParentProcessVsyncNotifier(unlabeled)
1.923s	1.840ms	Text	RefreshDriverTick - Tick reasons: HasObservers (DocumentTimeline animations [Style])
1.923s		NoPayloadU...	SetNeedStyleFlush
1.923s	176.00µs	Text	DoFlushPendingNotifications - Style
1.923s	165.80µs	Styles	Styles
1.923s	416.00ns	Styles	Styles
1.923s	342.46µs	Text	DoFlushPendingNotifications - InterruptibleLayout
1.923s	253.00ns	Styles	Styles
1.923s	194.00ns	Styles	Styles
1.923s	280.64µs	tracing	Reflow
1.923s	197.47µs	Text	RefreshObserver - Synthetic mouse move event [Display]
1.923s	120.58µs	tracing	DispatchSynthMouseMove
1.923s	3.762µs	Text	DoFlushPendingNotifications - InterruptibleLayout
1.923s	596.00ns	Styles	Styles
1.923s	244.00ns	Styles	Styles
1.924s	1.019ms	Text	Paint - Transaction ID: 369
1.924s	204.96µs	Text	Paint - DisplayList
1.924s	406.04µs	Text	Paint - WebRenderDisplayListBuilding
1.924s	38.391µs	Text	Paint - SendWebRenderTransactionToCompositor
1.925s	2.597µs	Text	Paint - Cleanup
1.925s	437.05µs	Text	Runnable - ProgressTracker::AsyncNotifyRunnable
1.925s	299.96µs	Text	Runnable - ProgressTracker::AsyncNotifyRunnable
1.925s	10.028µs	Text	Runnable - imgRequestProxy::RemoveFromLoadGroup
1.925s	2.024µs	Text	Runnable - imgRequestProxy::RemoveFromLoadGroup
1.925s	1.829µs	Text	Runnable - imgRequestProxy::RemoveFromLoadGroup
1.925s	4.583µs	Text	Runnable - PHal::Msg_UnlockScreenOrientation
1.925s	17.803µs	Text	Runnable - layers::ChromeProcessController::NotifyLayerTransforms
1.926s	1.921ms	Text	Runnable - PCompositorBridge::Msg_ObserveLayersUpdate
1.926s	1.887ms	DOMEvent	MozLayerTreeReady
1.926s		NoPayloadU...	SetNeedStyleFlush
1.926s	25.487µs	DOMEvent	beforeselect
1.926s	24.221µs	Text	DoFlushPendingNotifications - Style
1.926s	13.807µs	Styles	Styles
1.926s	435.00ns	Styles	Styles
1.926s		NoPayloadU...	SetNeedStyleFlush
1.927s		NoPayloadU...	SetNeedStyleFlush
1.927s		NoPayloadU...	SetNeedStyleFlush
1.927s		NoPayloadU...	SetNeedStyleFlush
1.927s		NoPayloadU...	SetNeedStyleFlush
1.927s		NoPayloadU...	SetNeedStyleFlush
1.927s		NoPayloadU...	SetNeedStyleFlush

608ns NotifyObservers

Details:

toggle-screenshot-disable

Thread:

Parent Process

Stack:

```
nsObserverService::NotifyObservers(n
NS_InvokeByIndex   XUL
XPCWrappedNative::CallMethod(XPCC
XPC_WN_CallMethod(JSContext*, unsig
0x20b313c9a767
onLocationChange   chrome://browser/
callListeners   chrome://browser/conte
_callProgressListeners   chrome://brow
Interpret(JSContext*, js::RunState&)   X
js::RunScript(JSContext*, js::RunState&
updateCurrentBrowser   chrome://brow
_setupEventListeners/<   chrome://brow
js::RunScript
js::InternalCallOrConstruct(JSContext*
js::Call(JSContext*, JS::Handle<JS::Val
JS::Call(JSContext*, JS::Handle<JS::Va
mozilla::dom::EventListener::HandleEv
mozilla::EventListenerManager::Handle
EventListener.handleEvent
mozilla::EventListenerManager::Handle
mozilla::EventTargetChainItem::Handle
mozilla::EventTargetChainItem::Handle
mozilla::EventDispatcher::Dispatch(nsI
EventDispatcher::Dispatch select
EventDispatcher::Dispatch
mozilla::EventDispatcher::DispatchDOM
nsINode::DispatchEvent(mozilla::dom::
mozilla::dom::EventTarget_Binding::dis
EventTarget.dispatchEvent
bool mozilla::dom::binding_detail::Gene
js::InternalCallOrConstruct(JSContext*
Interpret(JSContext*, js::RunState&)   X
js::RunScript(JSContext*, js::RunState&
set selectedIndex   chrome://global/co
js::RunScript
js::InternalCallOrConstruct(JSContext*
js::Call(JSContext*, JS::Handle<JS::Val
js::jit::InvokeFunction(JSContext*, JS::H
js::jit::InvokeFromInterpreterStub(JSCo
0x20b313c1fdf4
set selectedPanel   chrome://global/co
set selectedIndex   chrome://global/co
js::RunScript(JSContext*, js::RunState&
js::RunScript
js::InternalCallOrConstruct(JSContext*
js::CallSetter(JSContext*, JS::Handle<J
SetExistingProperty(JSContext*, JS::Ha
bool js::NativeSetProperty<(js::Qualifie
Interpret(JSContext*, js::RunState&)   X
js::RunScript(JSContext*, js::RunState&
set selectedItem   chrome://global/con
js::RunScript
js::InternalCallOrConstruct(JSContext*
js::CallSetter(JSContext*, JS::Handle<J
SetExistingProperty(JSContext*, JS::Ha
bool js::NativeSetProperty<(js::Qualifie
Interpret(JSContext*, js::RunState&)   X
js::RunScript(JSContext*, js::RunState&
set selectedTab   chrome://global/cont
js::RunScript
js::InternalCallOrConstruct(JSContext*
js::CallSetter(JSContext*, JS::Handle<J
SetExistingProperty(JSContext*, JS::Ha
bool js::NativeSetProperty<(js::Qualifie
Interpret(JSContext*, js::RunState&)   X
js::RunScript(JSContext*, js::RunState&
set selectedTab   chrome://browser/co
js::RunScript
js::InternalCallOrConstruct(JSContext*
js::CallSetter(JSContext*, JS::Handle<J
SetExistingProperty(JSContext*, JS::Ha
bool js::NativeSetProperty<(js::Qualifie
Interpret(JSContext*, js::RunState&)   X
js::RunScript(JSContext*, js::RunState&
_blurTab   chrome://b
```

■

- Gijs
  - Prooooootooooooon
- mconley

-
CAP'N CRUNCH'S
OOPS! All
PROTON
NATURALLY & ARTIFICIALLY FLAVORED
Sweetened Corn & Oat Cereal
- Bug 1704463 - 89.75 - 3.43% imgur ContentfulSpeedIndex / nytimes PerceptualSpeedIndex + 10 more (Linux, OSX, Windows) regression on Wed April 7 2021
  - TL;DR: The big regressions here seem to have been caused by a change in content viewport dimensions. This affects browsertime measurement, both in terms of the mathematics for doing visual metrics, but also changes how some sites operate due to what is visible on screen during the test.
- Helping mak with Bug 1704795 - 17.4 - 8.01% tart / tart + 3 more (Linux, OSX, Windows) regression on Sat April 10 2021
- Also looking at Bug 1706333 - 122.22 - 2.73% cnn-ampstories FirstVisualChange / ebay ContentfulSpeedIndex + 22 more (Windows) regression on Fri April 16 2021

Group Triage

-

Roundtable

- From last week:
  - [gijs] NPAPI plugin support is dead, long live NPA… actually no, let's keep it firmly where it is. But maybe there are perf improvements that are possible now all of that horridness is gone? I know there's some IPC that's now gone, and some ugly IO stuff that I wrote to try to make it not do mainthread IO, … there's probably other stuff!

April 13, 2021

Operations

- Next desktop meeting speaker is: dthayer

Engineering Updates

- bigiri
    -
- dthayer
    -
- emalysz
    -
- florian
    -
- Gijs
    -
- mconley
    -

Group Triage

-

Roundtable

- [gijs] NPAPI plugin support is dead, long live NPA… actually no, let’s keep it firmly where it is. But maybe there are perf improvements that are possible now all of that horridness is gone? I know there’s some IPC that’s now gone, and some ugly IO stuff that I wrote to try to make it not do mainthread IO, … there’s probably other stuff!

April 6th, 2021

Operations

- Next desktop meeting speaker is: dthayer

Engineering Updates

- bigiri
    -
- dthayer
    - Took a decent chunk of PTO
    - Proton stuff + debugging intermittents
    - Removed all references to the mostly non-functioning “Temporarily disable extension” button on the slow add-on notification
- emalysz
    - OS.File work wrapping up - all bugs are assigned and have patches
- florian
    - Not much to report. I spent time learning to use Pernosco, and debugging various test failures.
    - Success: filed a bug today for a new hang I found in BHR data. Emilio landed a fix within 4h!
- Gijs
    - Proton proton proton proton.
- mconley
    - Pre-XUL Skeleton UI experiment analysis completed
        - TL;DR: Bummer. :(No measurable difference in retention, days of use, or engagement. However, note the “Discussion” section regarding disproportionate unenrollment for people with slow disks.
            - *sad trombone* [dthayer]

Group Triage

-

Roundtable

-

March 30th, 2021 (Skipped due to low meeting week)

March 23rd, 2021

Operations

- Next desktop meeting speaker is: bigiri

Engineering Updates

- bigiri
    - Proton
- dthayer
    - Fixed a bug where the skeleton UI was breaking scroll inertia on Windows
    - Tracked down a UB issue originating in our usage of LZ4 from the startup cache. Turns out it's a problem deep in LZ4 itself. Emailed LZ4's original author, he says it's a known issue and doesn't have an intention of fixing it. This is a little bit troubling, and I'm still trying to work out a solution.
    - Proton stuff
- emalysz
    - Unblocked on OS.File bugs thanks to Barret landing the patch for Bug 1672431
- florian
    - I'm still working on fixing the tests that become perma-fail after removing the initial `TestUtils.waitForCondition` timer. At this point I'm down to 6 perma (or frequent) fails on try that I can't reproduce locally, where I intend to add a setTimeout so I can move on.
    - Next I'm hoping to have a look at making `BrowserTestUtils.switchTab` not take a minimum of 300ms.
    - I removed lots of MOZ_GECKO_PROFILER ifdefs in our tree. They are no longer needed to call functions like `profiler_is_active` `profiler_can_accept_markers`, … that are always declared.
    - I would like to make it easy to get profiles of a mochitest intermittent failure on try.
        - I encountered shutdown leaks on debug builds
    - [Fixed by Standard8] TelemetryController.jsm is now lazy loaded during desktop startup (and not loaded at all for Fenix).
- Gijs
    -
- mconley
    - Proton! Still!
    - Process priority manager is no longer being disabled with Fission, sounds like the Fission team is going to try to patch it to work with out-of-process subframes, because it seems to actually help with page load times on weaker hardware.
        - Lowering priority of ad processes: bug
        - Autokilling slow scripts in ad processes: bug
    -

Group Triage

-

Roundtable

-

March 16th, 2021

Operations

- Next desktop meeting speaker is: bigiri

Engineering Updates

- bigiri
    -
- dthayer
    - Implemented / improved UX of slow script notifications in Fission
    - Debugged the “Temporarily disable extension” button for add-on slow script notifications (turned out it has never worked - planning on just ripping it out)
    - Put patches up for Proton Skeleton UI (thanks Emma for reviews!)
    - Debugging a scrolling issue with touchpads and the skeleton UI (sometimes scrolling inertia doesn’t work - mysterious)
    - Proton stuff
- emalysz
    - Proton
    - Skeleton UI reviews
- florian
    - I added CSS transition and CSS animation markers: https://share.firefox.dev/2QaUjTK
    - “Categories with CPU” is now the default timeline view (front-end changes deployed today, and the ‘CPU’ feature was enabled by default yesterday on Nightly)
        - Nazim will demo tomorrow
    - I added many new markers in BrowserTestUtils and TestUtils to make it easy to figure out what a test is waiting for from watching a profile. Thanks Mike for the review!
    - I’m working on fixing all the test failures that happen when removing the initial 100ms timer that TestUtils.waitForCondition does. This makes some tests dramatically faster, and makes some existing intermittent failures permafails. Thanks Gijs for several quick reviews. (20 patches in the stack at this point, and at least 3 more test failures to investigate)
    - (and 2 proton patches)
- Gijs

    - Proton proton proton. Also proton. And some more proton.
- mconley
    - 100% Proton
    - Pre-XUL Skeleton UI experiment is halfway done. According to tdsmith, “The experiments are live and enrollment is looking good.”
    - Worked with k88hudson to confirm that Nimbus / ExperimentAPI.jsm would work perfectly for the about:home startup cache
    - Process priority manager is being disabled with Fission

Group Triage

-

Roundtable

-

March 9th, 2021 (Read-only)

Operations

- Next desktop meeting speaker is: bigiri
- This meeting is read-only due to a light meeting week.

Engineering Updates

- bigiri
    -
- dthayer
    -
- emalysz
    -
- florian
    -
- Gijs
    -
- mconley
    -

Group Triage

-

Roundtable

-

March 2nd, 2021

Operations

- Next desktop meeting speaker is: mconley
- Mike is on PTO today, so won't be able to let anyone in to his room…
    - We can use my room (REDACTED) instead.

Engineering Updates

- bigiri
    - Proton
- dthayer
    - Firedoge is back! (this has nothing to do with me - I just noticed it an am happy)
    - Slow script warning patches landed, and tentatively appear to have significantly reduced hang infobar reports (this is a good thing) - big thanks to Florian for reviewing these
    - Protonified the skeleton UI - don't have a patch up for this yet, but it's looking good locally
    - Lazy browser stuff appears to regress some things on talos, while improving other things. It's a bit of a christmas tree - trying to see if there's anything I can do about it
    - Working on Skeleton UI presentation
- emalysz
    - Proton
- florian
    - About:processes
        - Some UI fixes, thanks Gijs for the reviews!
        - Searched for nameless threads using lldb, Visual Studio and Pernosco. Identified a few, filed bugs, submitted some patches, got some fixed by the component teams.
    - I'm working on a patch so that when closing multiple tabs at once we run permitUnload in parallel and asynchronously on all tabs that have a beforeunload listener, before we start closing tabs without beforeunload listeners. I'm hoping this will significantly improve the perceived performance of closing many tabs.
- Gijs
    - Proton proton proton
    - Also florian's reviews!

- mconley
    - [bwinton] Surprise PTO today. Will fill this in later, but probably mostly Proton work.

Group Triage

-

Roundtable

-

February 23rd, 2021

Operations

- Next desktop meeting speaker is: mconley
-

Engineering Updates

- bigiri
    - Full time proton.
    - Had a build error that one other Git user experienced https://github.com/glandium/git-cinnabar/issues/267
- dthayer
    - Got roped into a Protonish slow script notification project.
        - Making the notification only show up when the user is trying to interact with a hung tab.
        - Blame the tab causing the hang
        - Other minor UX tweaks
    - Making _setupInitialBrowserAndTab run after first paint - I think try is finally green from this.
    - Toying with Fathom-related optimizations (6x improvement with a giant grain of salt)
- emalysz
    - Helped masterwayz land a patch that removes OS.File from XPIProvider.jsm (previously had been backed out)
    - As part of proton, we removed the slow startup notification
- florian
    - Reviewing Doug’s slow script patch.
    - Need to look more at about:processes, as Fission wants it:

        - Noticed that Mac thread names don’t need to be limited to 15 characters, and can go up to 63 characters. Sent patches to fix this on NSPR and JS code.
        - Started looking for threads that don’t have a name. I identified three so far using lldb to find a stack frame that would give me a hint about which code was running in the thread.
        - I will do some UI polish, and will need (a) reviewer(s)
            - Changing the layout from css display: table to display: grid so that long labels can overlap into the next cell when it’s empty.
            - Shortening wordings
            - Reducing l10n overhead
            - …
    - Added the “PROFILER” category for label frames, and used it to have nicer marker stacks that don’t show profiler internals
    - Cleaning up profiler includes in python-generated cpp files (ipc and webidl bindings).
    - Fixed some trivial perf bugs I found while profiling:
        - (still pending review) [Bug 1693877] Closing tabs trigger imports of FinderParent.jsm
        - (landed) [Bug 1693561] profiles.ini is stat'ed twice in a row during early startup
- Gijs
    - Proton proton proton
    - Made us not run permitUnload for hung tabs
- mconley
    - Working on Proton at roughly 100%
    - Pre-XUL / about:home startup experiment is now just the Pre-XUL experiment
        - Have QA sign-off now

Group Triage

- [mconley] Whoops

Roundtable

-

February 16th, 2021 (written only)

Operations

- Next desktop meeting speaker is: Gijs? Maybe? Did this meeting still happen?
- This meeting is written-only, due to a conflict with the Not Another BHM Panel event.

- **Don’t forget to fill out February 9th first (below), in case you missed that one.**

Engineering Updates

- bigiri
    -
- dthayer
    -
- emalysz
    -
- florian
    -
- Gijs
    -
- mconley
    -

Group Triage

-

Roundtable

-

February 9th, 2021 (written only)

Operations

- Next desktop meeting speaker is: Gijs? Maybe? Did this meeting still happen?
- This meeting is written-only, due to it being a low-meeting week.

Engineering Updates

- bigiri
    -
- dthayer
    -
- emalysz
    -
- florian
    -
- Gijs
    -

- mconley
    -

Group Triage

-

Roundtable

-

February 2nd, 2021

Operations

- Next desktop meeting speaker is: Gijs
- https://developer.mozilla.org/en-US/docs/Mozilla/Firefox/Performance_best_practices_for_Firefox_fe_engineers

Engineering Updates

- bigiri
    - More About dialog regressions, button text centering and missing bold in version number display.
- dthayer
    - Put together a proposal for a PreFirstPaint module - thank you to all who gave feedback!
    - Intending to dig into fixing whatever's going on with BHR data today
        - [florian] thanks!!
- emalysz
    - Working on a follow up patch to correctly display the bookmarks toolbar in the skeleton UI
    - Landed a patch that removes OS.File from DownloadIntegration.jsm that ensures we honor umask and adds a PathUtiils.toExtendedWindowsPath function
        - Exposed crashes in our makeDirectory implementation that Barret was able to fix
- florian
    - CPU data is now kept within the profile for profiles uploaded with the default profiler.firefox.com, the deploy preview is now only needed to view the data, not to capture it.
    - The better marker context menu I mentioned 2 weeks ago is now available on profiler.firefox.com

    - I added back the Sync IPC markers unconditionally (after bug 1557447 they were only recorded if the `ipcmessages` feature was set, which means we missed them in most profiles)
    - I’m experimenting with DoFlushPendingNotifications and SetNeedStyleFlush markers to make sync style flushes easy to understand. Example profile.
        - These markers don’t provide enough information to be useful without a stack, but capturing a stack is causing a lot of overhead, especially for a marker that can be added whenever the DOM is touched.
        - I tweaked the profiler code to collect Labels+JS stacks only (ie. no native stackwalking). In my measurements, adding a stackless marker takes about 1µs, adding a marker with a full stack takes about 40µs, and with my limited stacks about 20µs.
        - Trying to decide if the added value is enough to justify the additional overhead of these markers.
- Gijs
    - Didn’t do much, security issue + Proton took precedence.
    - Gave feedback on dthayer’s PreFirstPaint module proposal
    - Consulted with esmyth on unresponsive page document
- mconley
    - Gave feedback on dthayer’s PreFirstPaint module proposal
    - Consulted with esmyth on unresponsive page document
    - Did Lightning advisory for Pre-XUL Skeleton UI and about:home startup cache experiment
        - Filed the necessary JIRA tickets, got a number of sign-offs, waiting on Data Science sign-off
        - Reminder: we aim to run this experiment with 4 cohorts in Firefox 86 (Release) in early March.
            - Control cohort
            - Pre-XUL Skeleton UI + about:home startup cache
            - Pre-XUL Skeleton UI only
            - about:home startup cache only
    - Mainly working on Proton, and also on a series of hiring panels, so main fx-perf focus will be the experiment.

Product Updates

- esmyth
    - https://youtu.be/e7qQ6_RV4VQ?t=1

Group Triage

-

Roundtable

-

January 26th, 2021

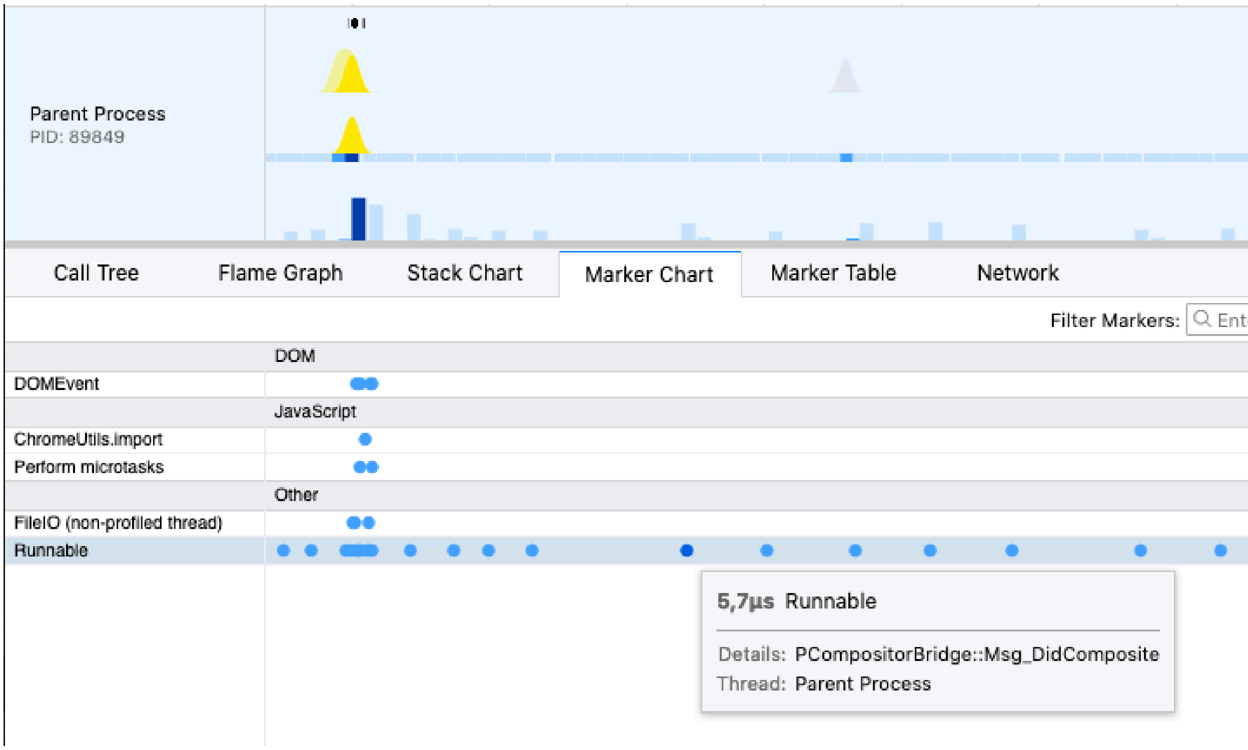
Operations

- Next desktop meeting speaker is: Gijs
- Instant-on Firefox
    - esmyth, if here, can maybe speak more about this
    - How could this work?
- Telemetry probes and sleeping
    - Bug 1205985 - Add Telemetry measurement for the time Firefox was open and the computer not in sleep
    - Bug 1685502 - Add a timestamp/duration class pair that only ticks when the computer is awake
- On MozAfterPaint usage
    - Archived doc which sadly did not survive the MDN migration. Mailing list post.

Engineering Updates

- bigiri
    - Trying to figure out how to “Refresh button spacing and style” for Proton.
- dthayer
    - tp5n/tp5o tests now load pages from representative domains
    - NS_NewNamedThread no longer blocks the main thread, assuming it sticks and we don’t notice anything off in the coming week(s)
        - [gijs] did this happen in 86? [dthayer] It did not.
    - Patch to get telemetry about skeleton UI disable reasons is landing right now, along with a patch to permanently(ish) disable it if we crash while showing it.
    - Working on this thing. (Left 4 startups are control, right 4 are with the patch) - Basically this is showing what happens if we don’t run a biggish chunk of JS before first paint. The question remains whether we can actually get away with this or not - the happy path of a default-configured Firefox with nothing weird on the system works great, but the code that we’re skipping is there to account for different circumstances.
- emalysz
    - Skeleton UI will now take into account the bookmarks toolbar
    - Patch that removes OS.File from DownloadIntegration.jsm was backed out for a windows failure related to Zone Identifier paths, so I’m working on implementing `winAllowLengthBeyondMaxPathWithCaveats` in PathUtils

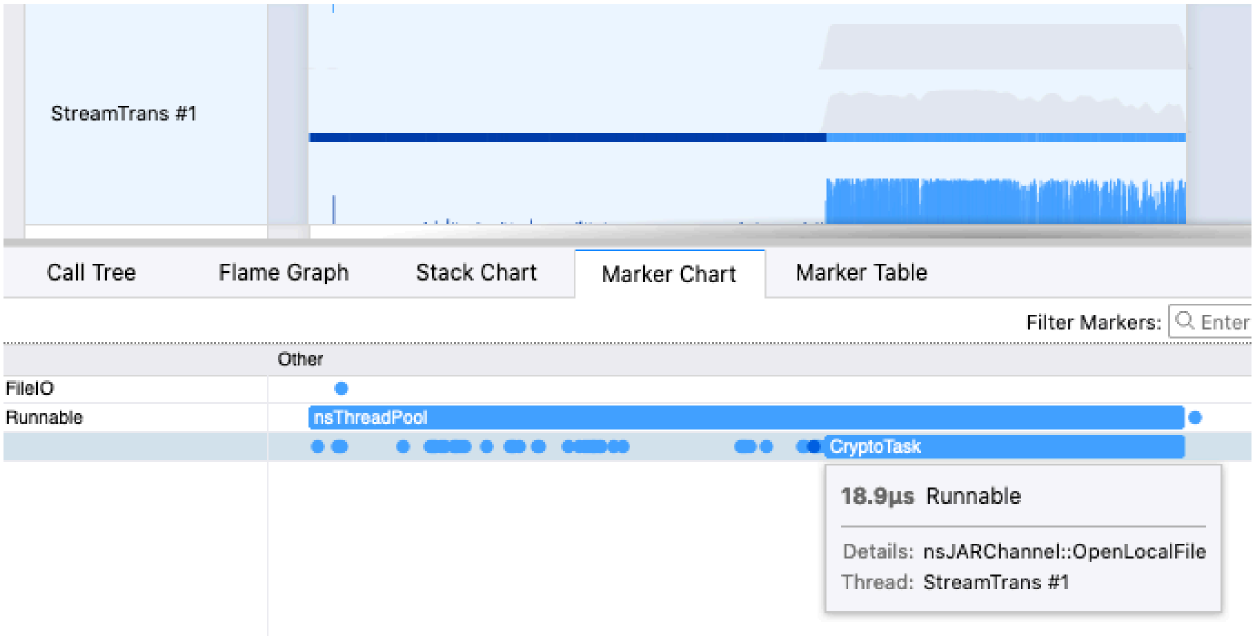
        - [gijs] I vaguely remember the security bug where this got added. LMK if you want to chat about it.
    - Worked on removing OS.File from JSONFile.jsm, but that looks like it's blocked by the AsyncShutdown blocker bug
        - [kmoir] I talked to Barret yesterday and he said he is pretty close to having a patch up for review for this blocker
- florian
    - Startup profiling and Talos profiling now collect CPU use data by default.
    - Mochitest and xpcshell test markers are now in the Test category.
    - I landed a new 'Runnable' marker that shows when a runnable is executed. For most runnable, it also shows a name.
        - This should make it possible to see why Firefox is using some CPU, but not enough to see it in the stack samples. Example:



5,7µs Runnable

Details: PCompositorBridge::Msg_DidComposite
Thread: Parent Process

        - This is not limited to the main thread. Ever wanted to know what's happening on StreamTrans threads?



18.9µs Runnable

Details: nsJARChannel::OpenLocalFile
Thread: StreamTrans #1

        - This shows “obvious” bugs, with messages being notified to processes or threads that don't care about them. Eg. vsync notified both the compositor and the main thread at once, when the main thread didn’t care.
    - Markus similarly landed a "Perform microtasks" marker.
- Gijs
    - Fiddled with bookmarks bar delays - now have telemetry and no longer delay initialization for new windows (to avoid appearing slow / hoping sqlite data is (disk) cached by then)
    - Fixed some issues with slow script telemetry
    - Still hoping to make some time for the DOM improvements - there’s some relatively low-hanging fruit, just not enough time in my days atm.
- mconley
    - Experimenter draft written, queued for Lightning Advisory for this Thursday
        - TL;DR: Going to try to do this on Release for 86.
    - Discussion with dthayer about Deuterium project

Product Updates

- esmyth
    - Proton investigation
        - Continued analysis of long-running script telemetry
            - Worked w/ Gijs on some telemetry issues
            - Working w/ Corey on some weirdness around active-ticks and hang duration
            - Interesting data

        - Aim to start slow startup analysis this week
    - Experiments
        - Fly on the wall for mconley's startup experiment(s)
        - Working to get moving on long-running script experiment
    - Mobile perf Q1 planning
        - Building a jupyter notebook for the android fe perf team to answer questions related to q1 planning and provide some detail on my methodology and tools for them to self-serve

Group Triage

- [gijs] Bookmark bar perf bug
    - Got a few ideas, just no time to implement. Happy to mentor someone who's not swamped with proton things?
        - mconley suggests bigiri or masterwayz

Roundtable

-

January 19th, 2021

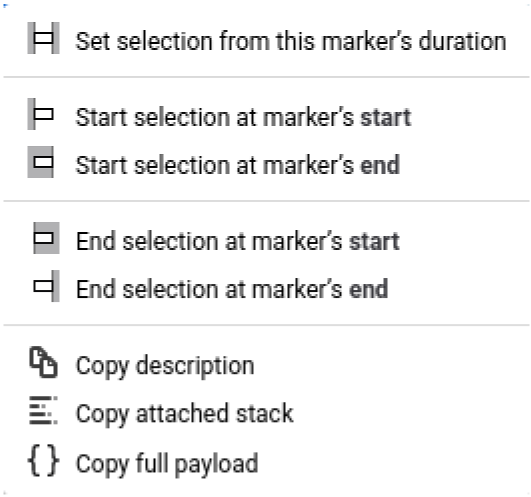
Operations

- Next desktop meeting speaker is: florian

Engineering Updates

- bigiri
    - About Dialog and IOUtils SharedDataMap patches have landed
    - There are two regressions in About Dialog, one about spacing between the links in the dialog and one about the links not working on Windows. The patches for both should be up soon.
- dthayer
    - Short week for me due to move / internet issues PTO.
    - Have a patch up for reporting skeleton UI failure reasons via telemetry, which also refactors a lot of the skeleton UI stuff to use mozilla::Results
- emalysz
    - Spent some time on polish bugs for Skeleton UI and have a patch up that will take into account the bookmarks toolbar

    - Patch up that removes OS.File from DownloadIntegration.jsm and changes IOUtils::SetPermissions to respect umask
    - Still looking into the snippets regression from converting SessionFile.jsm to IOUtils and why the SessionWorker/PromiseWorker patch causes a linux timeout
- florian
    - (coming soon) New context menu for markers (currently being reviewed)


Set selection from this marker's duration
Start selection at marker's start
Start selection at marker's end
End selection at marker's start
End selection at marker's end
Copy description
Copy attached stack
Copy full payload

- Gijs
    - Proton and reviews and stuff
    - Talked with esmyth about slow script warnings
        - Put up a bugfix for an issue Eric found with floats and decimal separators. TL;DR: you don't want l10n in your numerical telemetry metrics!
    - Bookmarks toolbar perf redux time!
        - Put up patches for bookmarks toolbar delays and measurements requested by esmyth (we might want placeholders in the toolbar for the delayed loading)
        - I broke some obscure way of doing bookmarks. Need to investigate.
    - Kicked off an email convo w/ kmag, mccr8 and Doug around the use of Cu.readUTF8/URLPreloader for JSONFile based on Emma's patch to switch that code to IOUtils.
    - For the lols: a performance optimization for fluent use with window titles broke with specific window titles because javascript is funky. (Fixed in 85 now!)
- mconley
    - Skeleton / about:home startup cache experiment
        - Having spoken to k88hudson, Nimbus isn't the right fit right now for this experiment, so going with Normandy / pref-flip experiment
        - Spent some time putting together this Experimenter doc for the experiment, still in draft phase while I sort a few things out, but hopefully will publish this week
    - Reviewed a patch to measure bookmark toolbar painting
    - Filed this bug about font loading at startup

    - Proton
        - There is interest in this project around improving the slow startup notification bar and slow script notification bar experience.

Product Updates

- esmyth
    - Proton investigation
    - Analyzing results of WarpBuilder experiment
        - General takeaway, we should avoid comparing means of client means. See this analysis from tdsmith for more
    - In progress analysis of long-running script telemetry
        - 11-12% of DAU experience at least one long-running script.
            - They don't necessarily see the warning. Earlier analysis showed ~½ aren't seen by the user. Need to confirm
        - Mirrored patterns for core active and regular users are interesting and need follow up
    - More discussions regarding perf telemetry/RUM
    - Mobile perf Q1 planning

Group Triage

-

Roundtable

January 12th, 2021

Operations

- Next desktop meeting speaker is: florian

Engineering Updates

- bigiri
    - IOUtils SharedDataMap is accepted and ready to land
    - About Dialog: I need to fix breaking changes in preferences due to re-used fluent messages.
- dthayer
    - Skeleton UI is enabled on Nightly by default. So far only one bug (associated with mochitests in headless mode) has been reported.

        - That being said, a few minor things have cropped up in my own use that I need to file bugs on and fix. So likely going to be spending a bit of time on that this week.
    - Put up a patch to have the tp5n and tp5o talos tests load pages from multiple domains, to get a little bit better coverage of changes which may introduce per-site overhead. This came up during the Network Partitioning performance consultation.
    - Looking into fixing bug 1640266 as a part of advancing fast shutdown on release. There’s no causal relationship, but given the possibility of perceived data loss during restarts it would be good to have if we wanted to announce the fast shutdown change in marketing updates.
- emalysz
    - Removed OS.File usage from ClientId.jsm and helped masterwayz with Sqlite.jsm
    - Patch up that removes OS.File from SessionWorker.jsm and changes how promise worker’s handle messages and errors
        - Intermittent shutdown timeout on Linux
        - Can land PageThumb.jsm patch once this lands
    - Working with masterwayz to change how we handle the profile extensions directory in the content process sandbox which should allow us to remove the makeDirectory call in XPIProvider
    - Working on a patch that lets us create a unique path in PathUtils
- florian
    - Compare CPU use for cold (almost no CPU used) and warm (CPU at 100% except during sync IPC) startup on the 2018 ref laptop.
    - CPU use recording on OS X is being fixed.
- Gijs
    - Filed a bunch of bugs around custom element / DOM overhead.
        - Need to file a few more.
        - Sort of interesting as “small” seeming overhead (1-5ms) is actually a substantial hit when opening windows / on our fast infra startup tests.
            - But hard to profile / hard to figure out how to make it faster short of “don’t do it”.
        - You can enable this instrumentation yourself using the MOZ_INSTRUMENT_CUSTOM_ELEMENTS environment variable. It’ll dump tables with metrics in the browser console. Note that as a gotcha, the totals are a naive sum across calls and thus can be overstated when there are nested calls (ie MyThing.foo calls into MyThing.bar), so be cautious about that when interpreting.
    -
- mconley

    - With the skeleton UI enabled in Nightly, I’m going to put together an Experimenter draft for running an experiment with skeleton / about:home startup cache enabled in Beta
        - Potential things to measure include:
            - Number of sessions (do users seem to be more comfortable restarting Firefox?)
            - Time to user interaction on about:home
            - Time to first page navigation
    - Mostly burning down Inbox (done), and Proton

Product Updates

- esmyth
    -

Group Triage

- (There was only bug this week, which mconley quickly triaged)

Roundtable

- https://twitter.com/FirefoxNightly/status/1349016425378230272

January 5th, 2021

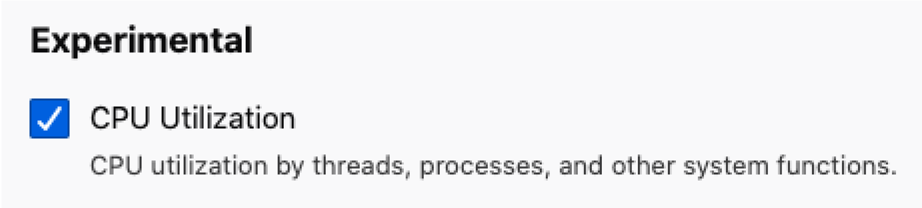
Operations

- Next desktop meeting speaker is: emalysz

Engineering Updates

- bigiri
    - IOUtils SharedDataMap
        - Found a bug after Kate’s approval and updated the patch
            - [gijs] I just re-reviewed this.
    - About Dialog
        - Addressed comments including a missed comment.
        - New bug introduced possibly due to adding support for checking if updates allowed in the policy before showing the updates. I will have to look more at the code to figure this out.

- dthayer
    - Propagated annotations from the BHR ping through the processing job, and modified Florian's BHR exploration tool to display annotation information. Need to send him the updated version to review and integrate.
    - Produced a few comparison videos between Chrome and Firefox illustrating some visual noise in common interactions.
        - Filed a few bugs to do with the address bar, referencing these.
    - Pulled down and analyzed late write stacks from telemetry in Beta, preparing to advance fast shutdown to 3 on Beta/Release. It looks more or less clean.
    - The Pref flip patch for enabling the skeleton UI on Nightly is r+'d by mhowell - giving it a little bit to see if anyone else on the list has concerns and then planning on landing it.
- emalysz
    - Worked on RTL support for the Skeleton UI
    - Have a patch up that replaces OS.File.Error handling in promise workers and removes OS.File usage from SessionWorker
        - Hopefully will unblock a few of the other promise worker bugs
    - Hoping to take a look at 2 patches from masterwayz that were backed out for causing regressions when switching to IOUtils
        - XPIProvider.jsm
        - SessionFile.jsm: regressions >150% on tp5n nonmain_startup_fileio, >50% on tp5n main_startup_fileio, sessionrestore
- florian
    - Experimental "CPU Utilization" profiler feature has landed today:

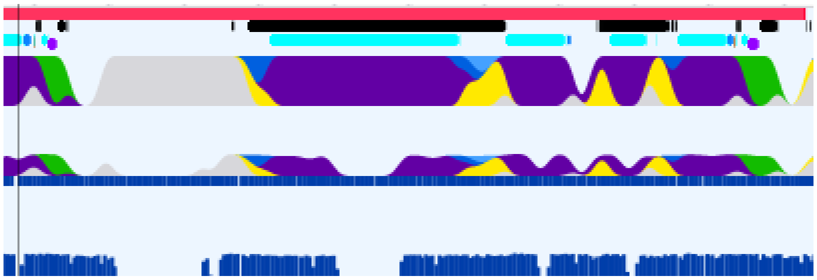
      

      Experimental

      ☑ CPU Utilization

      CPU utilization by threads, processes, and other system functions.

      Set **devtools.performance.recording.ui-base-url** to https://deploy-preview-3098--perf-html.netlify.app in about:config to see the data.
      This is all still experimental.
      Example profile:
      https://deploy-preview-3098--perf-html.netlify.app/public/zfhekrk18sbex9nxv4qsqfs7mn2v97es11a50yr/calltree/?globalTrackOrder=7-0-1-2-3-4-5-6&hiddenGlobalTracks=1-2-3-4-5&hiddenLocalTracksByPid=41428-0&localTrackOrderByPid=41428-2-0-1~41429-0~41433-0~41437-0~41432-0~41430-0~41431-0~&range=1624m2224~2721m747&thread=0&v=5



- Gijs
  - Digging out from PTO.
  - Need to deal with some Proton stuff, and some download-related regressions on release.
  - Did some DOM stuff before PTO. Some interesting results.
- mconley
  - Back from PTO today, digging myself out from my inbox.
  - Needinfos seem to have really piled up, too. Expect minor delays this week.

Product Updates

- esmyth
  - Task 1, figure out what I was working on before PTO. Task 2, finish setting up macbook
  - Catching up on a few things: email, OKR, mdn developer needs survey, some Q4 UR, web vision, desktop operating vision
  - Dusting off WIP research: slow script, ysod, slow startup, bhr

Group Triage

-

Roundtable

-

Archives

- May 7, 2019 - December 15, 2020
- September 18, 2018 - April 30, 2019
- Jan 12, 2018 - September 11, 2018