Firefox

Firefox power profiling

A powerful visualization of web sustainability

Link to slides:
https://share.firefox.dev/power-profiling-fosdem2024

Florian Quèze

February 4, 2024 - Brussels

Web sustainability

The carbon footprint of browsing the web comes from:

- the user device
- remote equipment (network, servers, ...)
- power used by the browser on the device

The user device

Embodied emissions of the device (laptop, smartphone, …)

- We don’t pick the user’s device…
- … but can reduce incentives to replace it
    - Performance
    - Compatibility with old devices

      Firefox ESR 115 is the only browser receiving security updates that still supports Windows 7

Infrastructure emissions

Caused by the resources fetched by the web page.

- The financial cost (mostly) scales with emissions
- Good incentive to reduce it.
- Already a lot of tools to look at web page optimization from this perspective

Browser power use

Often neglected because hard to measure:

- There’s no good tooling available
- … except now there is!

  Focus for the rest of this talk

Table of contents

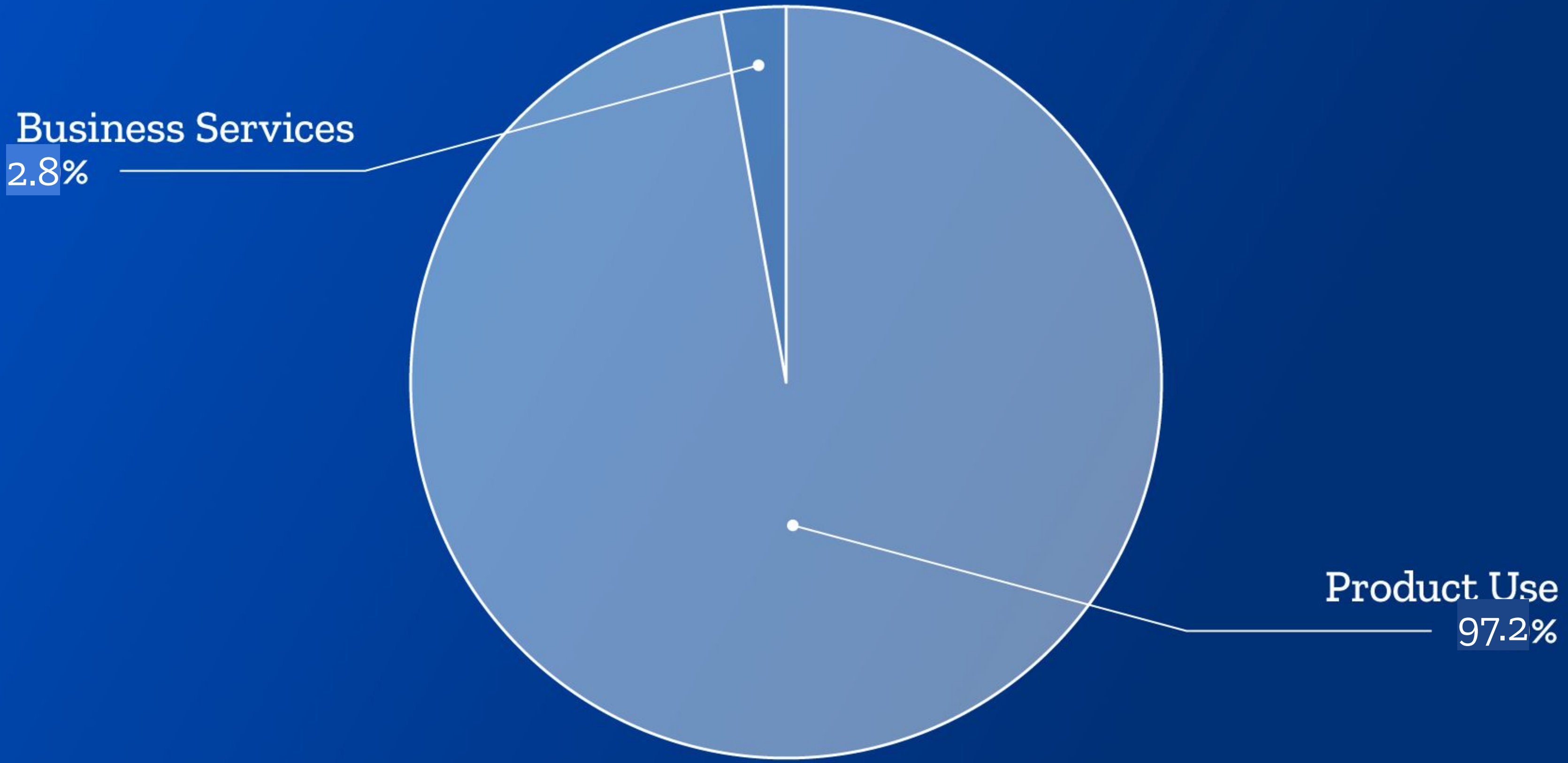
Why?

For sustainability

Mozilla made climate commitments:

- being carbon-neutral.
- reducing its GHG footprint year over year
- **leading openly by sharing materials, tools, and methodologies**.
- exploring approaches to develop, design, and **improve products from a sustainability perspective**

Emissions Distribution 2022



For our users

Excessive power use = poor user experience

- Noisy fans
- Hot laptops
- Short battery life

For a better web

Mozilla’s mission:

“***We’re building a better Internet***”

Tools created to make Firefox more energy efficient are directly re-usable to make web pages more efficient.

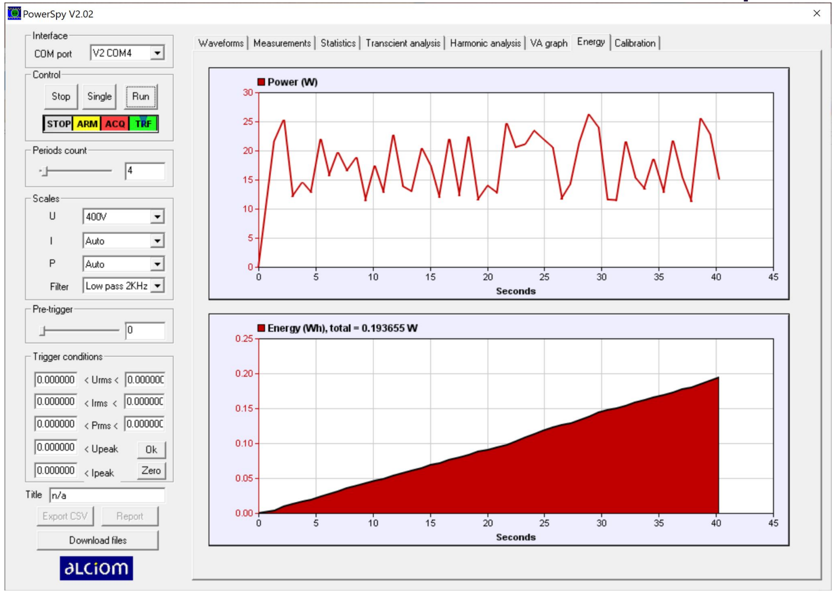
Measuring local power use

(cheap) Wattmeter

- Affordable
- Reasonably accurate
- Can’t track evolution over time

Better wattmeter

- Connects over bluetooth
- Data readable from another computer



PowerSpy V2.02

Interface
COM port: V2 COM4

Control
Stop | Single | Run
STOP | ARM | ACQ | TRF

Periods count: 4

Scales
U: 400V
I: Auto
P: Auto
Filter: Low pass 2KHz

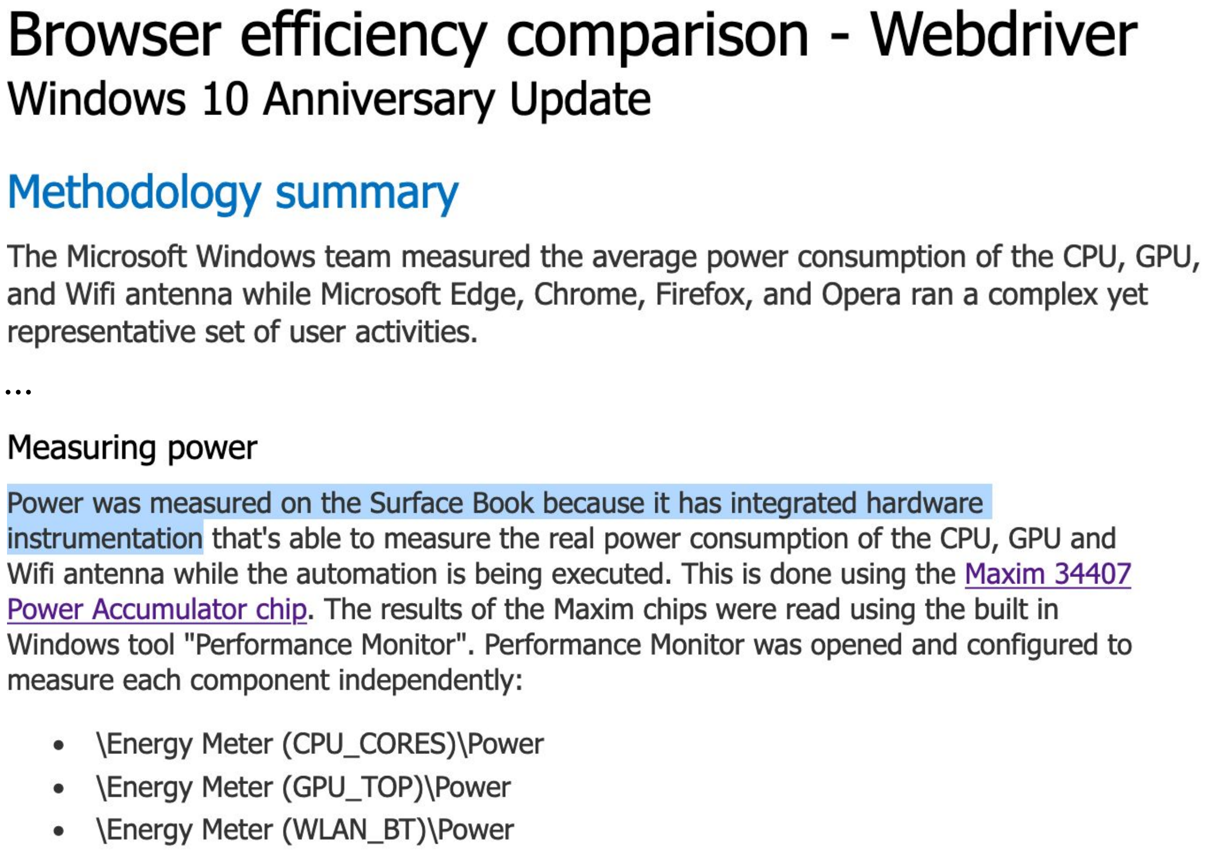
Pre-trigger: 0

Trigger conditions
0.000000 < Urms < 0.000000
0.000000 < Irms < 0.000000
0.000000 < Prms < 0.000000
0.000000 < Upeak
0.000000 < Ipeak

Title: n/a

- Still difficult to match with what was done

How did Microsoft do it?



Browser efficiency comparison - Webdriver

Windows 10 Anniversary Update

Methodology summary

The Microsoft Windows team measured the average power consumption of the CPU, GPU, and Wifi antenna while Microsoft Edge, Chrome, Firefox, and Opera ran a complex yet representative set of user activities.

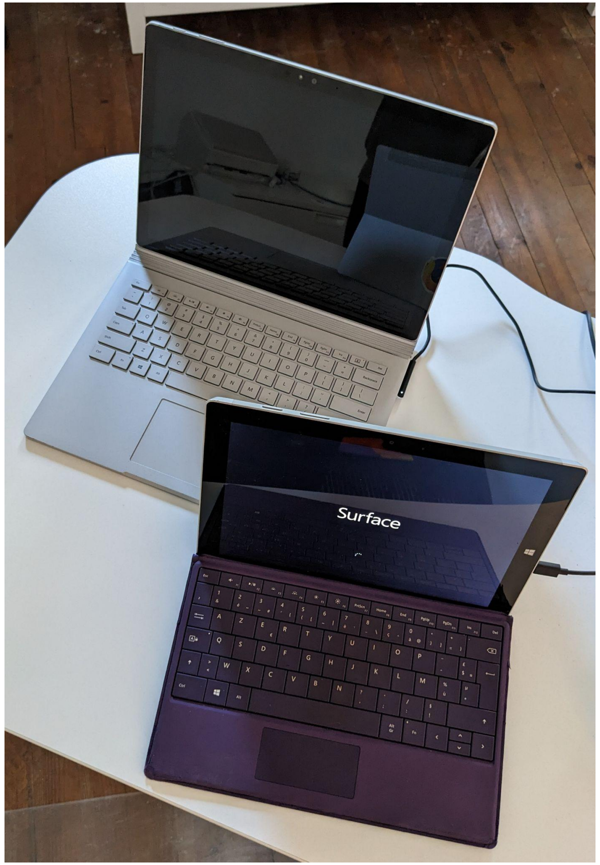
...

Measuring power

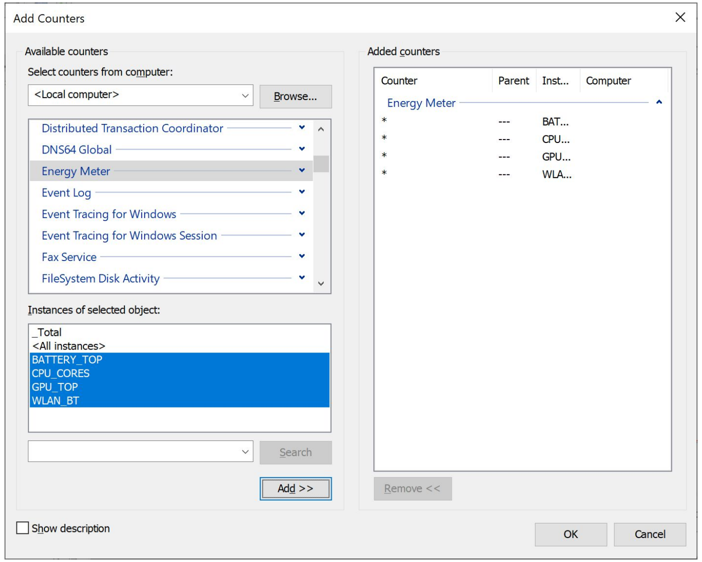
Power was measured on the Surface Book because it has integrated hardware instrumentation that's able to measure the real power consumption of the CPU, GPU and Wifi antenna while the automation is being executed. This is done using the Maxim 34407 Power Accumulator chip. The results of the Maxim chips were read using the built in Windows tool "Performance Monitor". Performance Monitor was opened and configured to measure each component independently:

- \Energy Meter (CPU_CORES)\Power
- \Energy Meter (GPU_TOP)\Power
- \Energy Meter (WLAN_BT)\Power

Devices with MAXIM power meters



- Surface Book 1, Surface 3
- Many other devices expose it in their ACPI table, but don’t actually have the chip
- Energy meters in perfmon.exe:



Add Counters

Available counters

Select counters from computer:

<Local computer>

Browse...

Distributed Transaction Coordinator
DNS64 Global
Energy Meter
Event Log
Event Tracing for Windows
Event Tracing for Windows Session
Fax Service
FileSystem Disk Activity

Instances of selected object:

_Total
<All instances>
BATTERY_TOP
CPU_CORES
GPU_TOP
WLAN_BT

Search

Add >>

Added counters

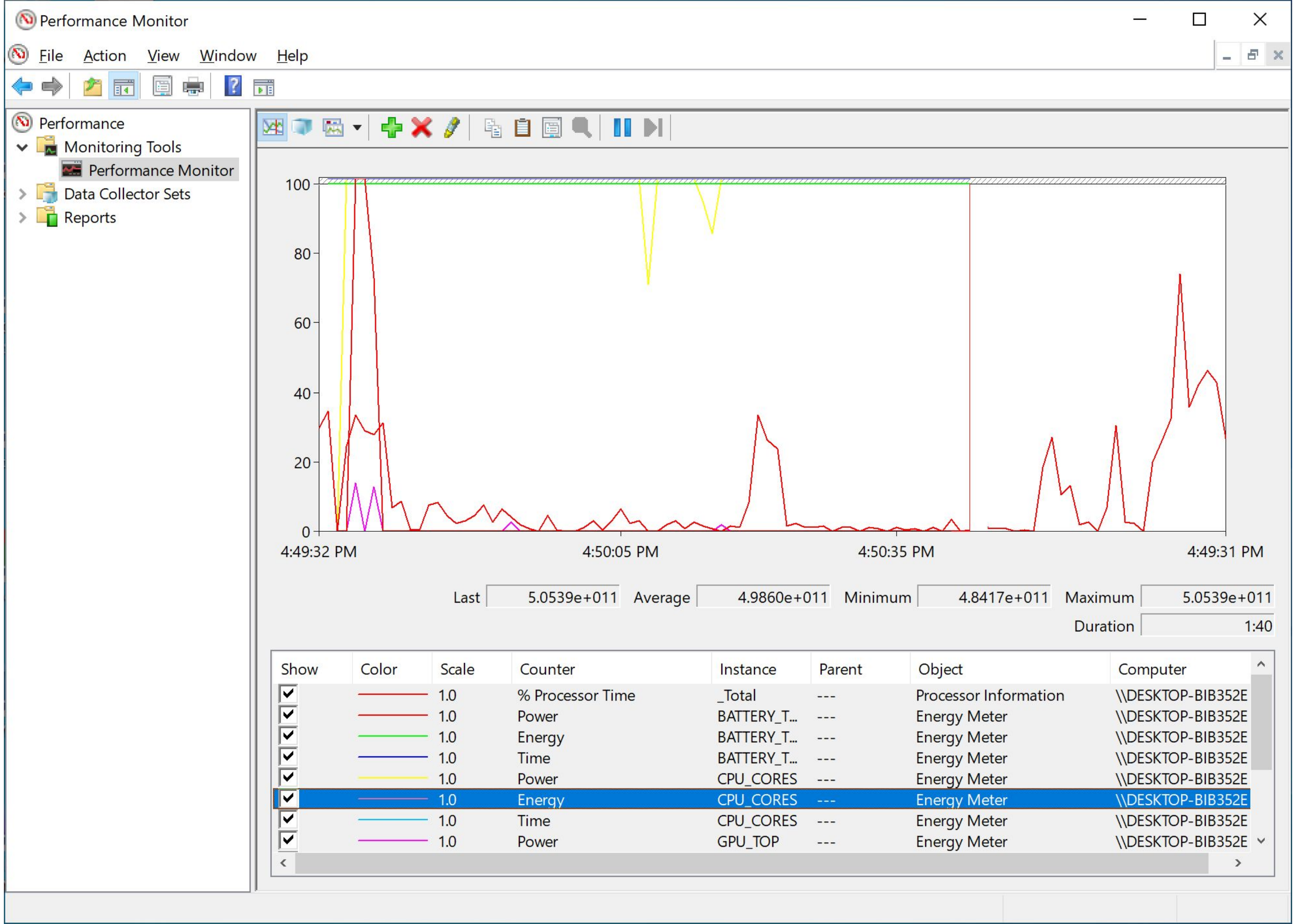
Counter	Parent	Inst...	Computer
Energy Meter			
*	---	BAT...	
*	---	CPU...	
*	---	GPU...	
*	---	WLA...	

Remove <<

Show description

OK

Cancel

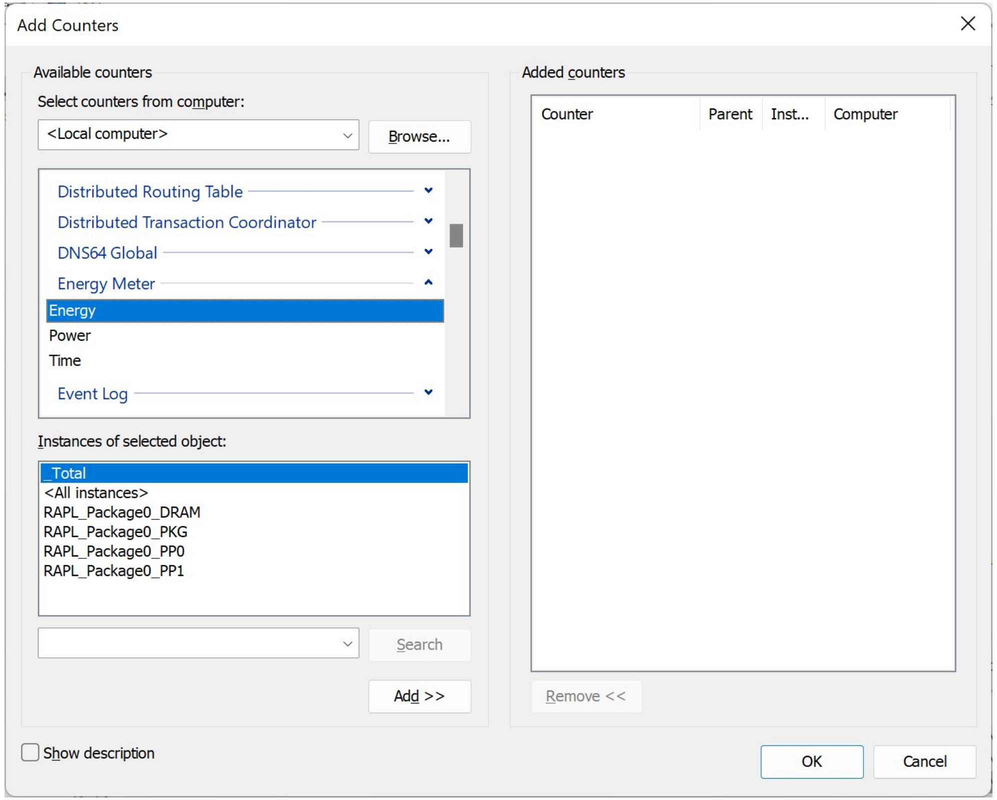


Last	Average	Minimum	Maximum
5.0539e+011	4.9860e+011	4.8417e+011	5.0539e+011

Duration 1:40

Show	Color	Scale	Counter	Instance	Parent	Object	Computer
✓	🟥	1.0	% Processor Time	_Total	---	Processor Information	\\DESKTOP-BIB352E
✓	🟥	1.0	Power	BATTERY_T...	---	Energy Meter	\\DESKTOP-BIB352E
✓	🟩	1.0	Energy	BATTERY_T...	---	Energy Meter	\\DESKTOP-BIB352E
✓	🟦	1.0	Time	BATTERY_T...	---	Energy Meter	\\DESKTOP-BIB352E
✓	🟨	1.0	Power	CPU_CORES	---	Energy Meter	\\DESKTOP-BIB352E
✓	🟪	1.0	Energy	CPU_CORES	---	Energy Meter	\\DESKTOP-BIB352E
✓	🟦	1.0	Time	CPU_CORES	---	Energy Meter	\\DESKTOP-BIB352E
✓	🟪	1.0	Power	GPU_TOP	---	Energy Meter	\\DESKTOP-BIB352E

A good surprise...


Add Counters

Available counters

Select counters from computer:

<Local computer>

Distributed Routing Table
Distributed Transaction Coordinator
DNS64 Global
Energy Meter
Energy
Power
Time
Event Log

Instances of selected object:

_Total
<All instances>
RAPL_Package0_DRAM
RAPL_Package0_PKG
RAPL_Package0_PP0
RAPL_Package0_PP1

Added counters

Counter	Parent	Inst...	Computer

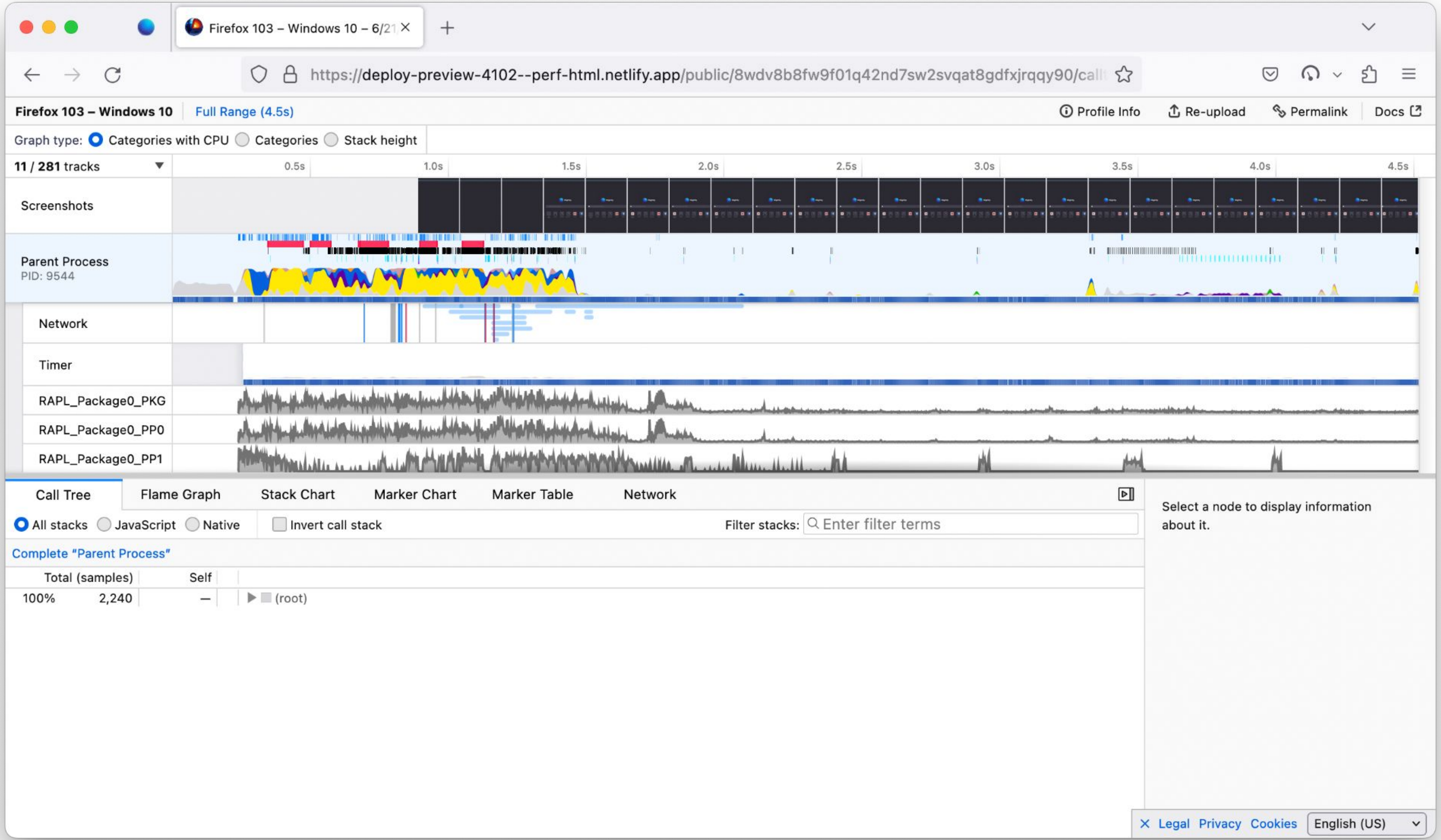
- Some machines report energy meter channels with familiar names.
- Windows 11 with Intel CPUs.

Energy Meter Interface API

- The perfmon.exe UI is horrible, but...
- There's a documented API!
    - unit is picowatt-hour
    - can be queried many times per second
    - accessible in user land

      (no requirement to install a specific driver)
- Usable for profiler counters:

  Bug 1774844 - Use the Windows Energy Meter Interface to record power use data in profiles



Firefox 103 – Windows 10 – 6/21

https://deploy-preview-4102--perf-html.netlify.app/public/8wdv8b8fw9f01q42nd7sw2svqat8gdfxjrqqy90/call

Firefox 103 – Windows 10 | Full Range (4.5s)

Profile Info | Re-upload | Permalink | Docs

Graph type: [✓ Categories with CPU] Categories Stack height

11 / 281 tracks

Call Tree | Flame Graph | Stack Chart | Marker Chart | Marker Table | Network

[✓ All stacks] JavaScript Native | Invert call stack

Filter stacks: Enter filter terms

Complete "Parent Process"

Total (samples)		Self	
100%	2,240	—	▶ (root)

Select a node to display information about it.

Legal Privacy Cookies English (US)

Introducing the Firefox Profiler

https://profiler.firefox.com

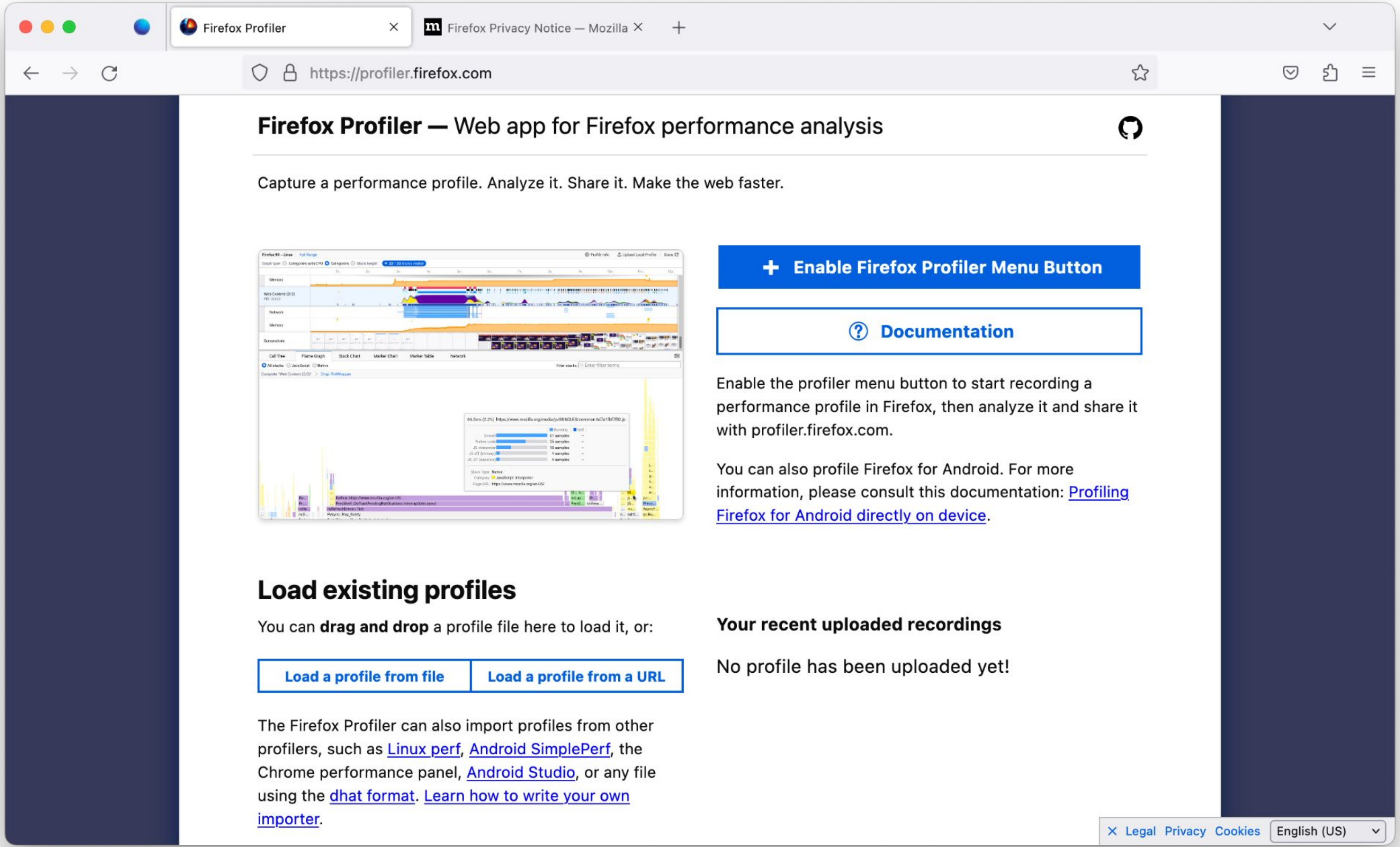
What’s the Firefox Profiler

- Built-into Firefox
- Initially for performance work. Helps:
  - users to make useful bug reports
  - developers to make sense of them
- One of the best profilers!

What's the Firefox Profiler

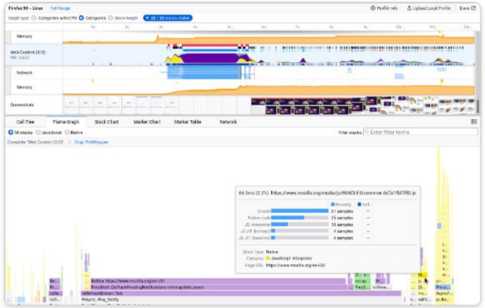
Main sources of data:

- Sampling
  - Stacks
  - Counters
- Markers


Firefox Profiler
Firefox Privacy Notice — Mozilla
https://profiler.firefox.com

Firefox Profiler — Web app for Firefox performance analysis

Capture a performance profile. Analyze it. Share it. Make the web faster.



+ Enable Firefox Profiler Menu Button

Documentation

Enable the profiler menu button to start recording a performance profile in Firefox, then analyze it and share it with profiler.firefox.com.

You can also profile Firefox for Android. For more information, please consult this documentation: Profiling Firefox for Android directly on device.

Load existing profiles

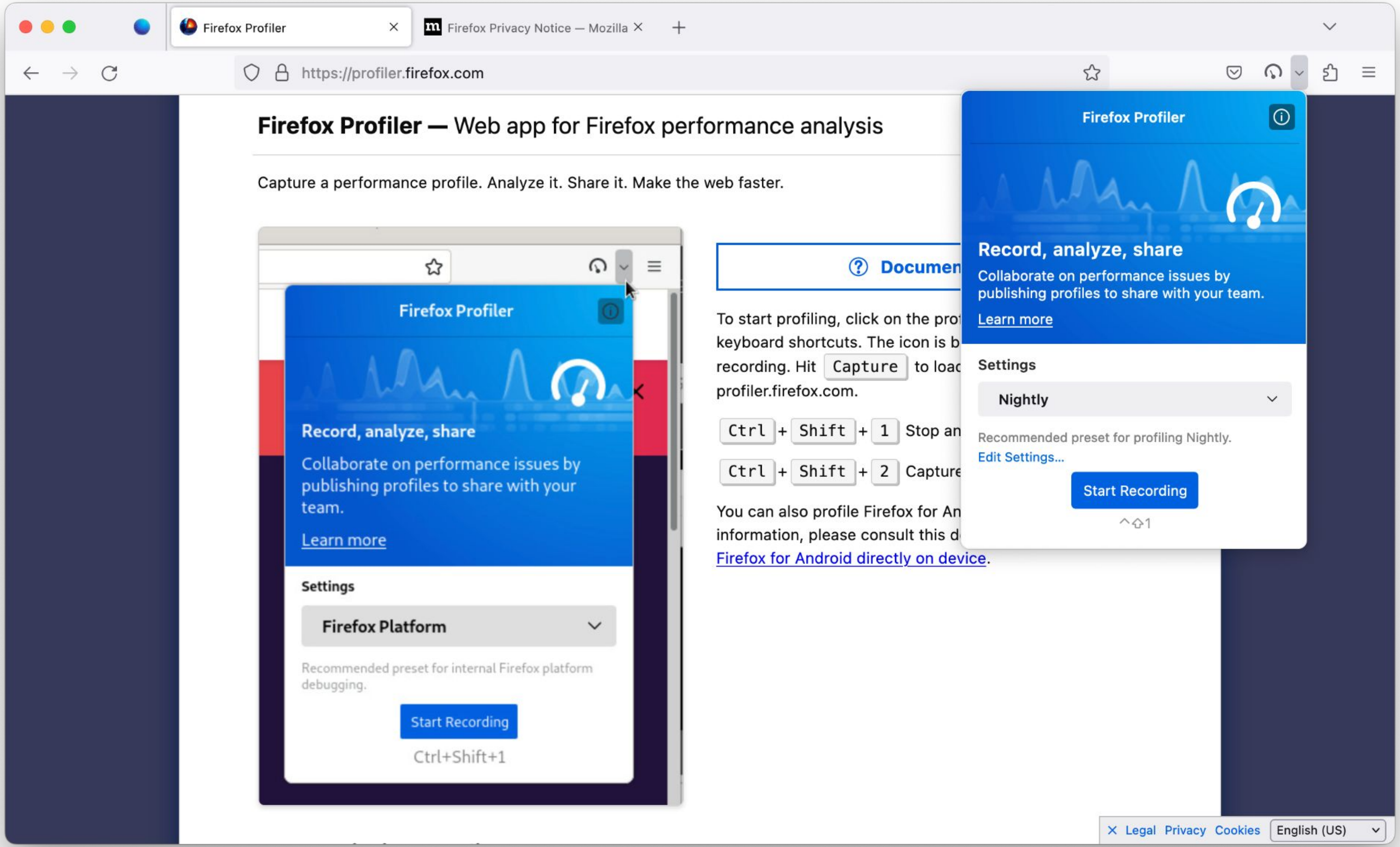
You can **drag and drop** a profile file here to load it, or:

Load a profile from file | **Load a profile from a URL**

The Firefox Profiler can also import profiles from other profilers, such as Linux perf, Android SimplePerf, the Chrome performance panel, Android Studio, or any file using the dhat format. Learn how to write your own importer.

Your recent uploaded recordings

No profile has been uploaded yet!


Firefox Profiler

Firefox Privacy Notice — Mozilla

https://profiler.firefox.com

Firefox Profiler — Web app for Firefox performance analysis

Capture a performance profile. Analyze it. Share it. Make the web faster.

Documen

To start profiling, click on the prof keyboard shortcuts. The icon is b recording. Hit `Capture` to load profiler.firefox.com.

`Ctrl` + `Shift` + `1` Stop an

`Ctrl` + `Shift` + `2` Capture

You can also profile Firefox for An information, please consult this d Firefox for Android directly on device.

Firefox Profiler

Record, analyze, share

Collaborate on performance issues by publishing profiles to share with your team.

Learn more

Settings

Nightly

Recommended preset for profiling Nightly.

Edit Settings...

Start Recording

^⇧1

Firefox Profiler

Record, analyze, share

Collaborate on performance issues by publishing profiles to share with your team.

Learn more

Settings

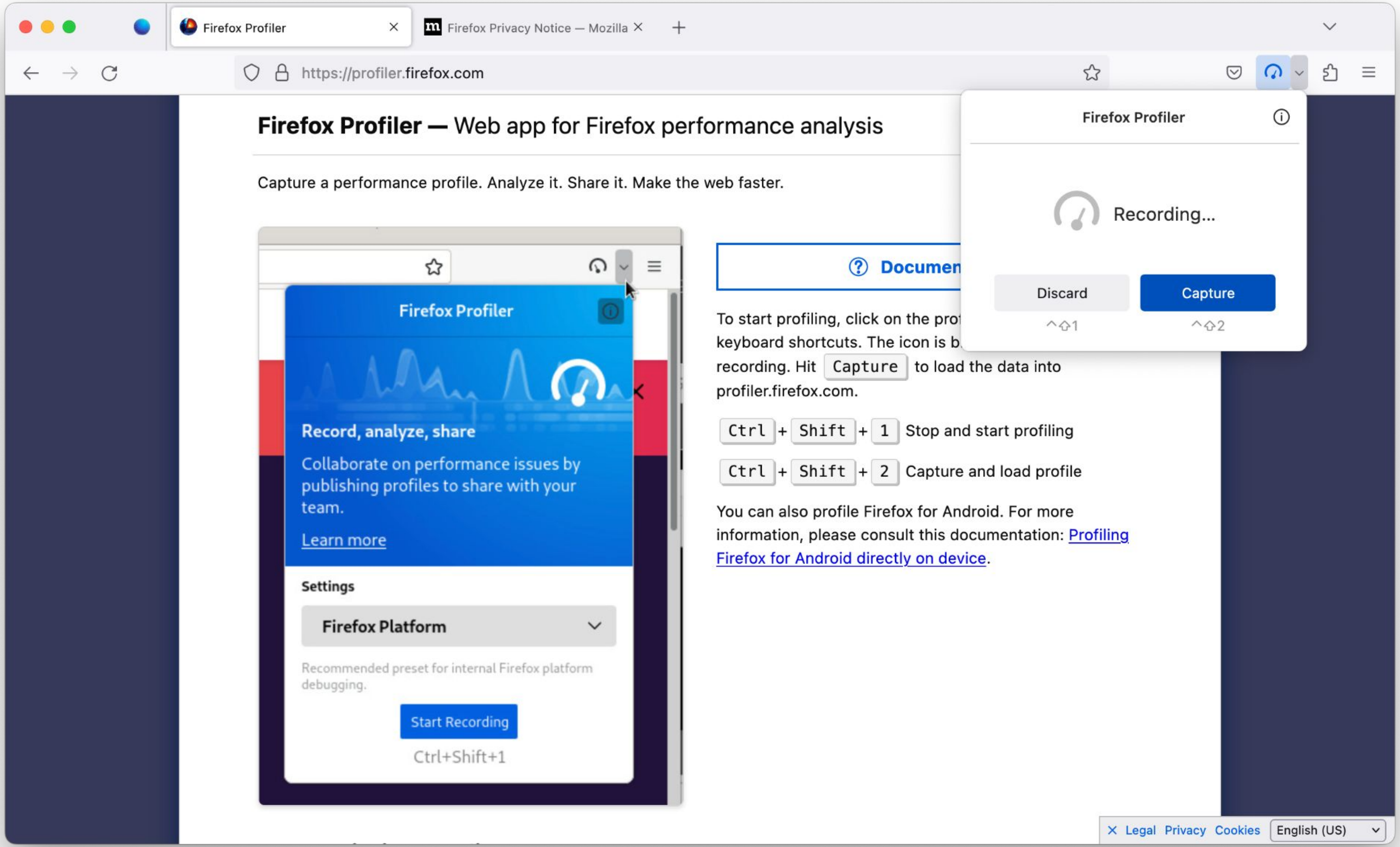
Firefox Platform

Recommended preset for internal Firefox platform debugging.

Start Recording

Ctrl+Shift+1

Legal Privacy Cookies English (US)


Firefox Profiler
Firefox Privacy Notice — Mozilla
https://profiler.firefox.com

Firefox Profiler — Web app for Firefox performance analysis

Capture a performance profile. Analyze it. Share it. Make the web faster.

Collaborate on performance issues by publishing profiles to share with your team.

Firefox Platform

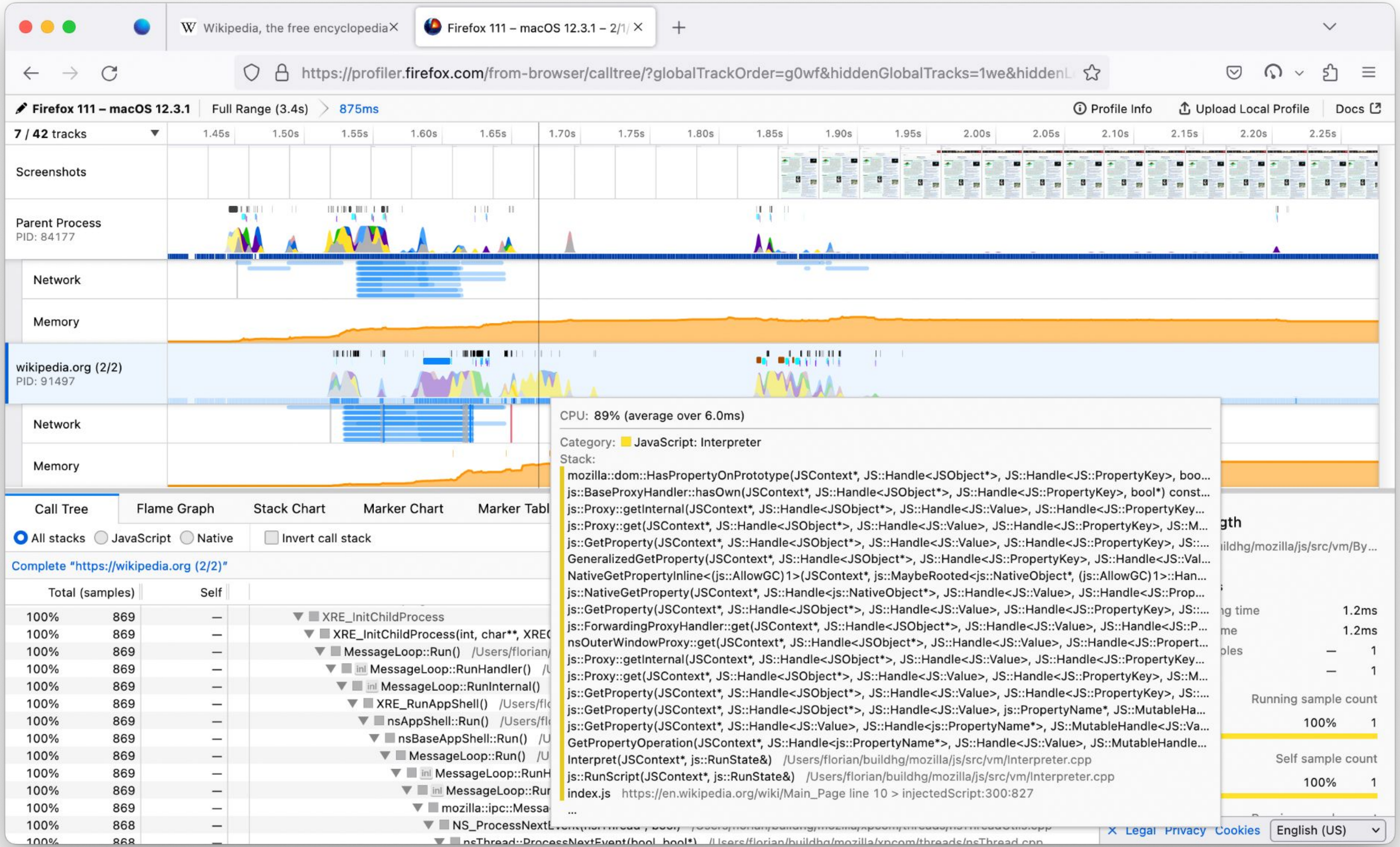
Recommended preset for internal Firefox platform debugging.

To start profiling, click on the prof keyboard shortcuts. The icon is b recording. Hit `Capture` to load the data into profiler.firefox.com.

`Ctrl` + `Shift` + `1` Stop and start profiling

`Ctrl` + `Shift` + `2` Capture and load profile

You can also profile Firefox for Android. For more information, please consult this documentation: Profiling Firefox for Android directly on device.



CPU: 89% (average over 6.0ms)

Category: JavaScript: Interpreter

Stack:

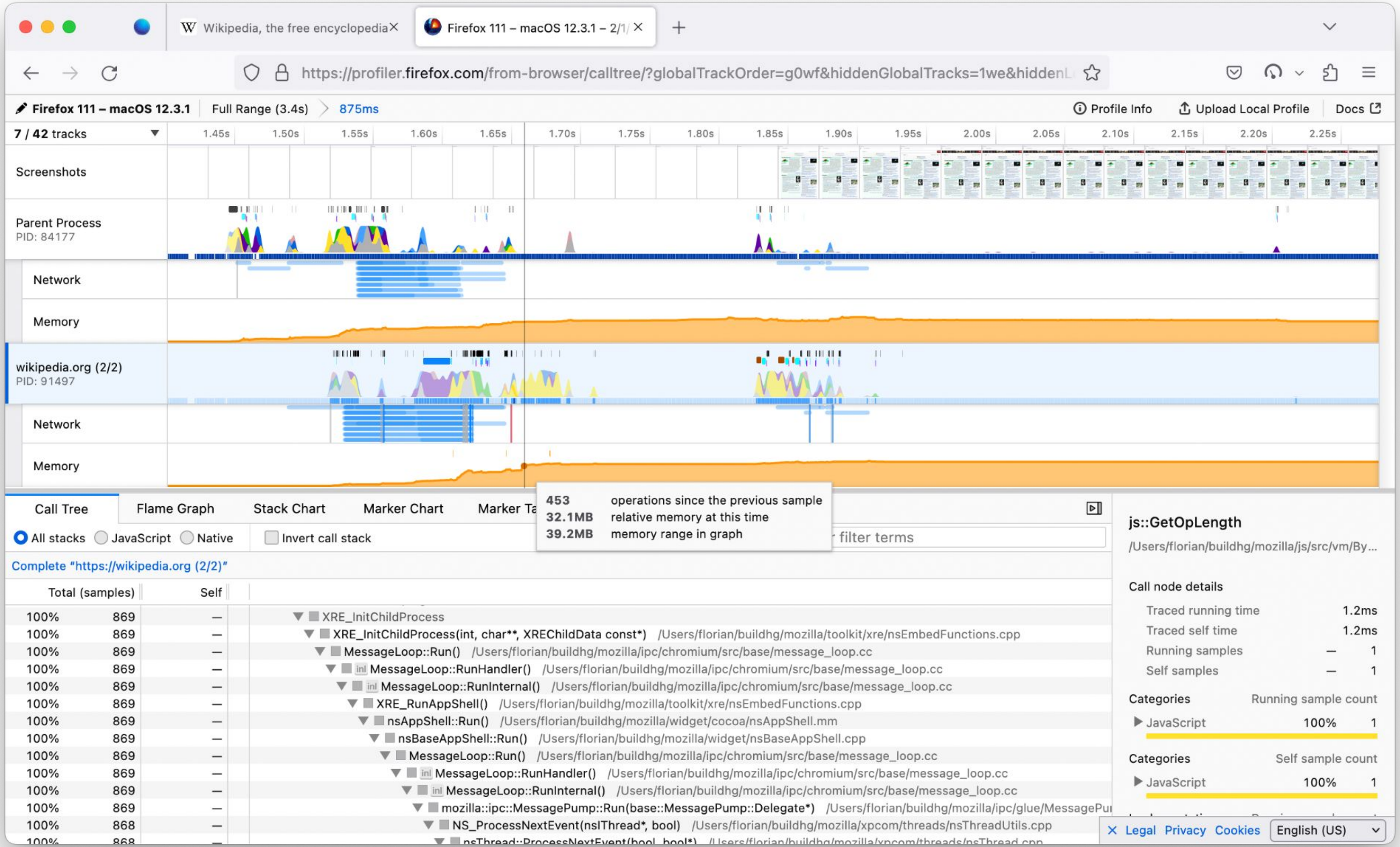
```
mozilla::dom::HasPropertyOnPrototype(JSContext*, JS::Handle<JSObject*>, JS::Handle<JS::PropertyKey>, boo...
js::BaseProxyHandler::hasOwn(JSContext*, JS::Handle<JSObject*>, JS::Handle<JS::PropertyKey>, bool*) const...
js::Proxy::getInternal(JSContext*, JS::Handle<JSObject*>, JS::Handle<JS::Value>, JS::Handle<JS::PropertyKey...
js::Proxy::get(JSContext*, JS::Handle<JSObject*>, JS::Handle<JS::Value>, JS::Handle<JS::PropertyKey>, JS::M...
js::GetProperty(JSContext*, JS::Handle<JSObject*>, JS::Handle<JS::Value>, JS::Handle<JS::PropertyKey>, JS::...
GeneralizedGetProperty(JSContext*, JS::Handle<JSObject*>, JS::Handle<JS::PropertyKey>, JS::Handle<JS::Val...
NativeGetPropertyInline<(js::AllowGC)1>(JSContext*, js::MaybeRooted<js::NativeObject*, (js::AllowGC)1>::Han...
js::NativeGetProperty(JSContext*, JS::Handle<js::NativeObject*>, JS::Handle<JS::Value>, JS::Handle<JS::Prop...
js::GetProperty(JSContext*, JS::Handle<JSObject*>, JS::Handle<JS::Value>, JS::Handle<JS::PropertyKey>, JS::...
js::ForwardingProxyHandler::get(JSContext*, JS::Handle<JSObject*>, JS::Handle<JS::Value>, JS::Handle<JS::P...
nsOuterWindowProxy::get(JSContext*, JS::Handle<JSObject*>, JS::Handle<JS::Value>, JS::Handle<JS::Propert...
js::Proxy::getInternal(JSContext*, JS::Handle<JSObject*>, JS::Handle<JS::Value>, JS::Handle<JS::PropertyKey...
js::Proxy::get(JSContext*, JS::Handle<JSObject*>, JS::Handle<JS::Value>, JS::Handle<JS::PropertyKey>, JS::M...
js::GetProperty(JSContext*, JS::Handle<JSObject*>, JS::Handle<JS::Value>, JS::Handle<JS::PropertyKey>, JS::...
js::GetProperty(JSContext*, JS::Handle<JSObject*>, JS::Handle<JS::Value>, js::PropertyName*, JS::MutableHa...
js::GetProperty(JSContext*, JS::Handle<JS::Value>, JS::Handle<js::PropertyName*>, JS::MutableHandle<JS::Va...
GetPropertyOperation(JSContext*, JS::Handle<js::PropertyName*>, JS::Handle<JS::Value>, JS::MutableHandle...
Interpret(JSContext*, js::RunState&)  /Users/florian/buildhg/mozilla/js/src/vm/Interpreter.cpp
js::RunScript(JSContext*, js::RunState&)  /Users/florian/buildhg/mozilla/js/src/vm/Interpreter.cpp
index.js  https://en.wikipedia.org/wiki/Main_Page line 10 > injectedScript:300:827
...
```

Total (samples)		Self	
100%	869	—	XRE_InitChildProcess
100%	869	—	XRE_InitChildProcess(int, char**, XREC
100%	869	—	MessageLoop::Run() /Users/florian/
100%	869	—	MessageLoop::RunHandler() /U
100%	869	—	MessageLoop::RunInternal()
100%	869	—	XRE_RunAppShell() /Users/flo
100%	869	—	nsAppShell::Run() /Users/flo
100%	869	—	nsBaseAppShell::Run() /U
100%	869	—	MessageLoop::Run() /U
100%	869	—	MessageLoop::RunH
100%	869	—	MessageLoop::Rur
100%	869	—	mozilla::ipc::Messa
100%	868	—	NS_ProcessNextEvent(nsIThread*, bool) /Users/florian/buildhg/mozilla/xpcom/threads/nsThreadUtils.cpp
100%	868	—	nsThread::ProcessNextEvent(bool, bool*) /Users/florian/buildhg/mozilla/xpcom/threads/nsThread.cpp

gth		
ildhg/mozilla/js/src/vm/By...		
ng time		1.2ms
me		1.2ms
ples	—	1
	—	1
Running sample count		
	100%	1
Self sample count		
	100%	1

X Legal Privacy Cookies English (US)

Loading the wikipedia home page - https://share.firefox.dev/3I5H1aF



453	operations since the previous sample
32.1MB	relative memory at this time
39.2MB	memory range in graph

Complete "https://wikipedia.org (2/2)"

Total (samples)		Self	
100%	869	—	XRE_InitChildProcess
100%	869	—	XRE_InitChildProcess(int, char**, XREChildData const*) /Users/florian/buildhg/mozilla/toolkit/xre/nsEmbedFunctions.cpp
100%	869	—	MessageLoop::Run() /Users/florian/buildhg/mozilla/ipc/chromium/src/base/message_loop.cc
100%	869	—	inl MessageLoop::RunHandler() /Users/florian/buildhg/mozilla/ipc/chromium/src/base/message_loop.cc
100%	869	—	inl MessageLoop::RunInternal() /Users/florian/buildhg/mozilla/ipc/chromium/src/base/message_loop.cc
100%	869	—	XRE_RunAppShell() /Users/florian/buildhg/mozilla/toolkit/xre/nsEmbedFunctions.cpp
100%	869	—	nsAppShell::Run() /Users/florian/buildhg/mozilla/widget/cocoa/nsAppShell.mm
100%	869	—	nsBaseAppShell::Run() /Users/florian/buildhg/mozilla/widget/nsBaseAppShell.cpp
100%	869	—	MessageLoop::Run() /Users/florian/buildhg/mozilla/ipc/chromium/src/base/message_loop.cc
100%	869	—	inl MessageLoop::RunHandler() /Users/florian/buildhg/mozilla/ipc/chromium/src/base/message_loop.cc
100%	869	—	inl MessageLoop::RunInternal() /Users/florian/buildhg/mozilla/ipc/chromium/src/base/message_loop.cc
100%	869	—	mozilla::ipc::MessagePump::Run(base::MessagePump::Delegate*) /Users/florian/buildhg/mozilla/ipc/glue/MessagePu
100%	868	—	NS_ProcessNextEvent(nsIThread*, bool) /Users/florian/buildhg/mozilla/xpcom/threads/nsThreadUtils.cpp
100%	868	—	nsThread::ProcessNextEvent(bool, bool*) /Users/florian/buildhg/mozilla/xpcom/threads/nsThread.cpp

js::GetOpLength
/Users/florian/buildhg/mozilla/js/src/vm/By...

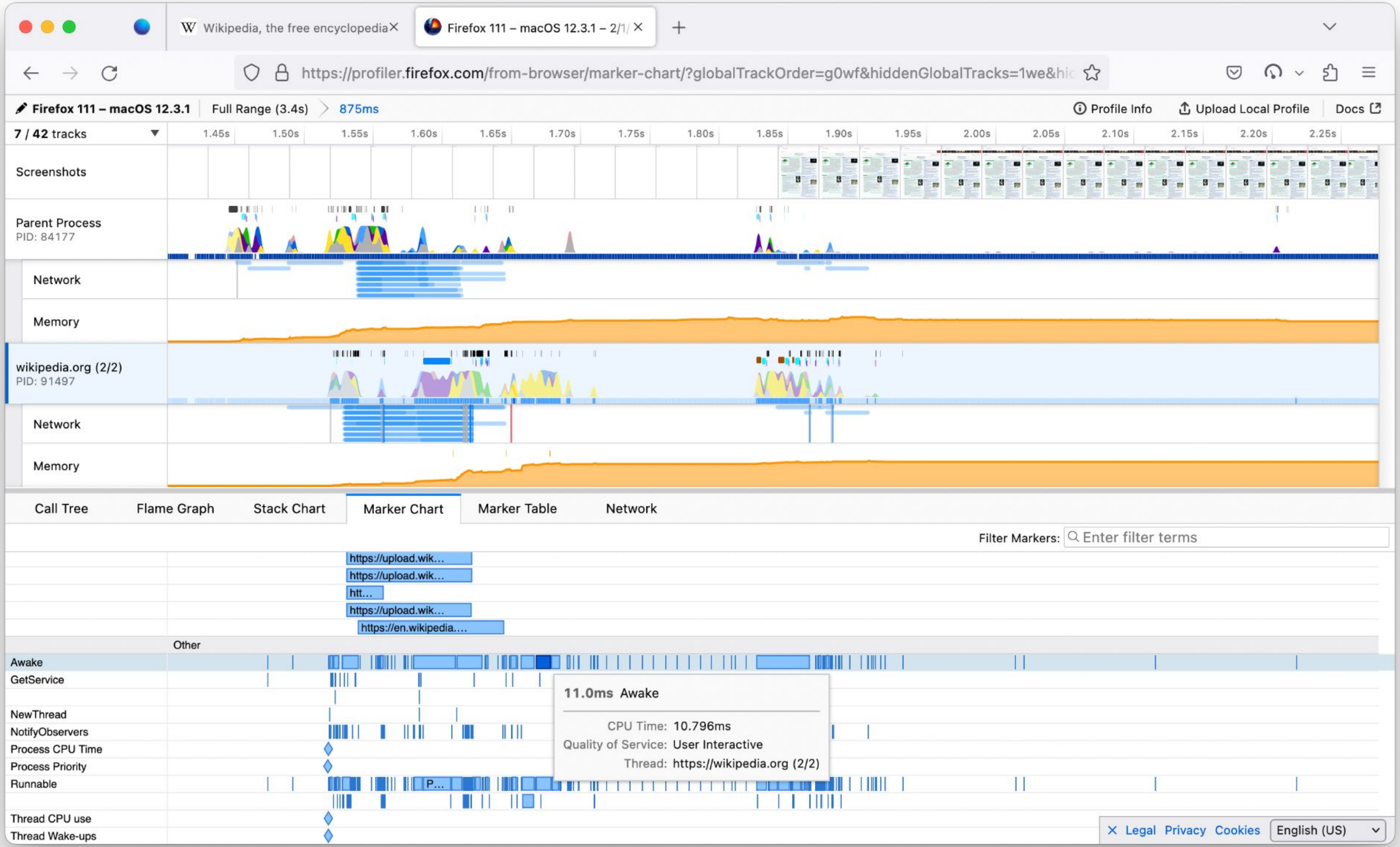
Call node details

Traced running time		1.2ms
Traced self time		1.2ms
Running samples	—	1
Self samples	—	1

Categories	Running sample count	
JavaScript	100%	1

Categories	Self sample count	
JavaScript	100%	1

Loading the wikipedia home page - https://share.firefox.dev/3I5H1aF



```
11.0ms  Awake

         CPU Time: 10.796ms
Quality of Service: User Interactive
           Thread: https://wikipedia.org (2/2)
```

Loading the wikipedia home page - https://share.firefox.dev/3l5H1aF

Firefox power profiling

Firefox Power profiling

- Built-in, no extra tool required
- Supports all 3 major desktop platforms
- Shipped in Firefox 104 (June 2022)
- Not copied yet!

Power profiling - Windows support

- Windows 10 - devices with hardware power meters
  (Surface Book 1, Surface 3, ?)
  CPU, GPU, Wifi power use
- Windows 11 - Intel CPUs
  CPU, GPU, DRAM power use
- Windows 11 22H2 - AMD Ryzen CPUs
  CPU, with 1 track per core!

Power profiling - Mac support

- Apple Silicon
  - Undocumented API, returning a per-process value!

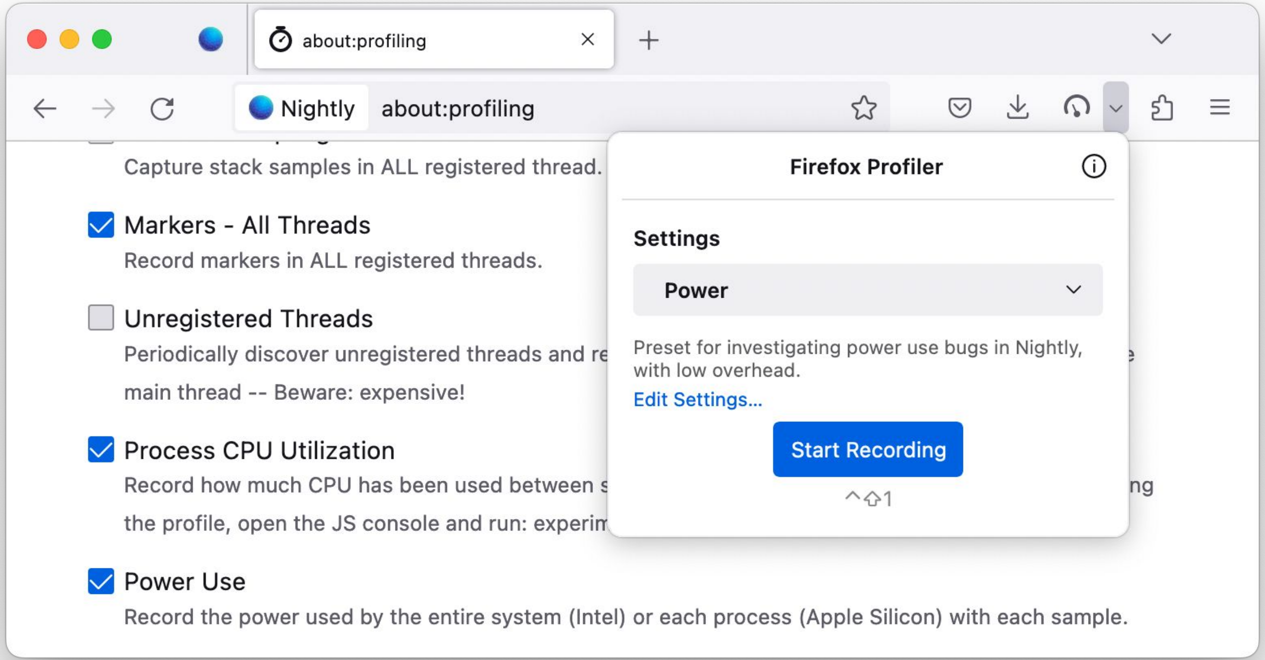
    ```
    task_info(mach_task_self(), TASK_POWER_INFO_V2,
              (task_info_t)&task_power_info, &count);
    task_power_info.task_energy //  ← nanojoules
    ```
- Intel x64-64 CPUs
  - `diagCall64(dgPowerStat,` … called from asm gives us the RAPL MSR.
  - (copied from an 9 years old implementation)

Power profiling - Linux support

- Use RAPL perf events
- `sudo sysctl kernel.perf_event_paranoid=0`
  Access to power data is restricted since October 2020 due to a side channel attack.
- AMD CPUs supported since Linux Kernel 5.8
- Doesn’t work with Ubuntu’s Firefox snap package

Power profiling - configuration

There is a ‘Power’ preset for easy configuration.



Capture stack samples in ALL registered thread.

✓ Markers - All Threads
Record markers in ALL registered threads.

☐ Unregistered Threads
Periodically discover unregistered threads and re
main thread -- Beware: expensive!

✓ Process CPU Utilization
Record how much CPU has been used between s
the profile, open the JS console and run: experim

✓ Power Use
Record the power used by the entire system (Intel) or each process (Apple Silicon) with each sample.

Firefox Profiler

Settings

Power

Preset for investigating power use bugs in Nightly, with low overhead.

Edit Settings…

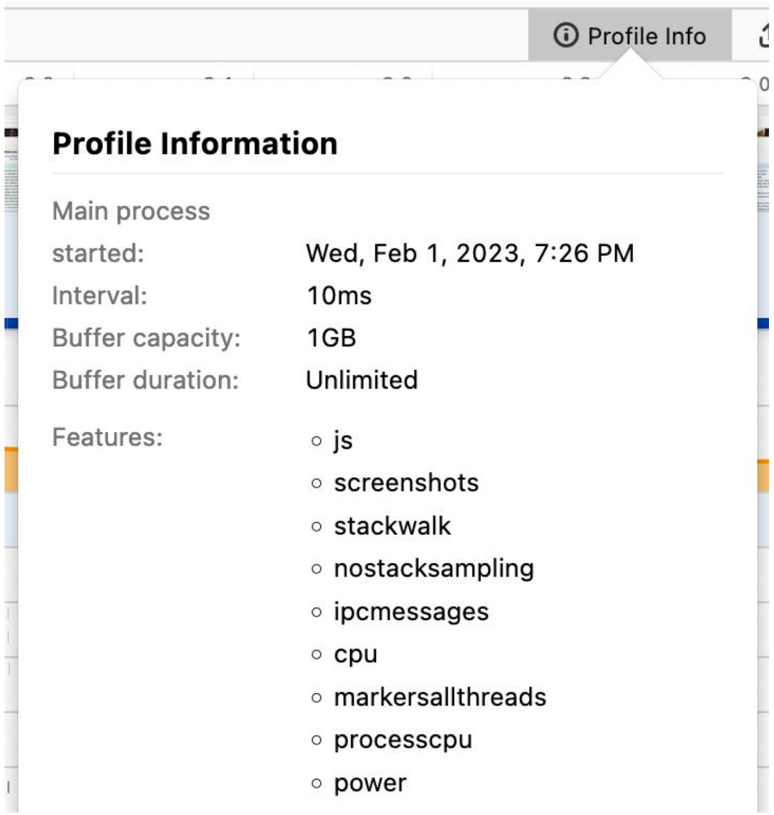
Start Recording

^⇧1

Windows 11 & Apple Silicon since Firefox 104 — Linux & Intel Macs since 107

Reducing overhead

- Longer interval
- No periodic stack sampling



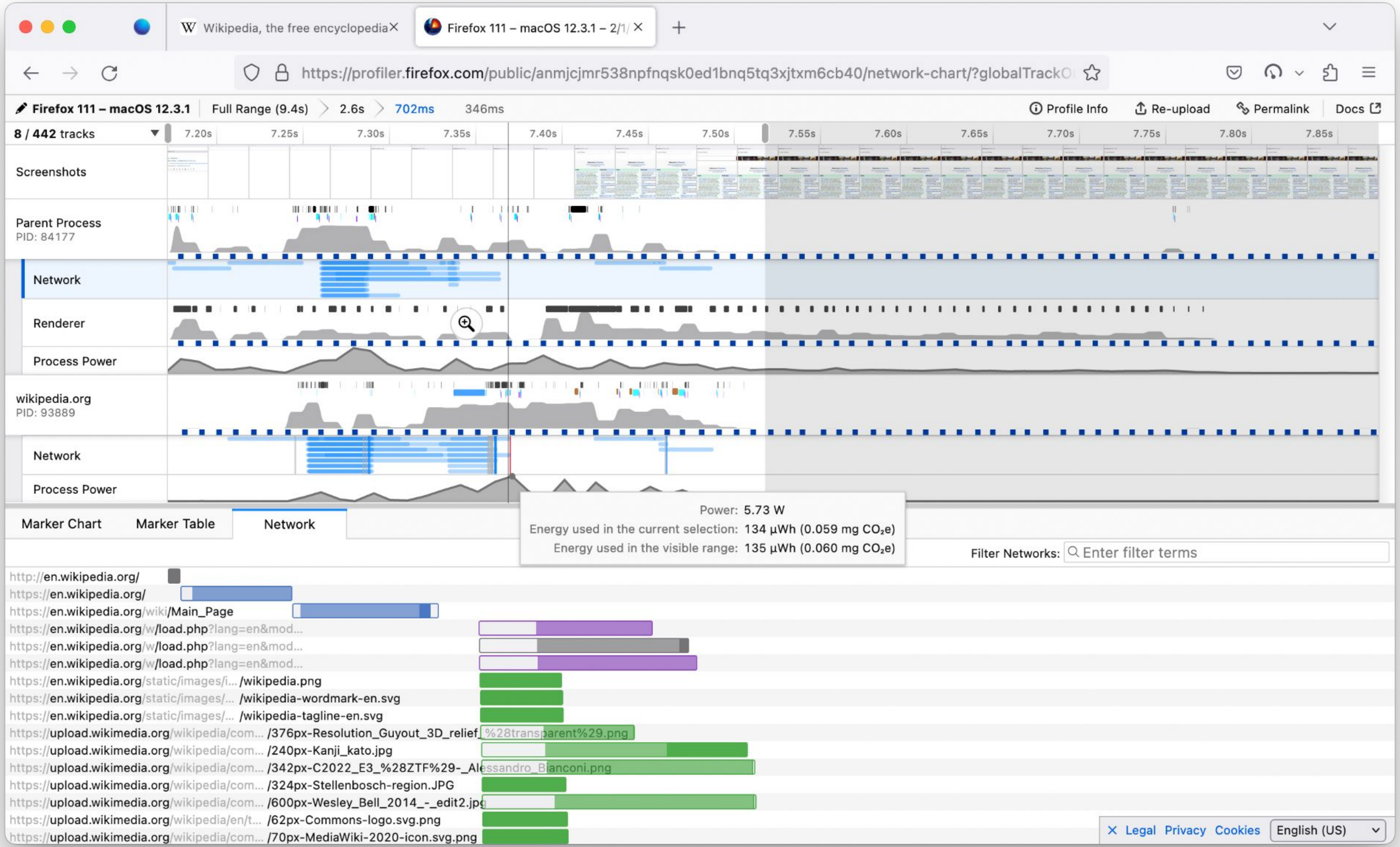
Profile Information

Main process

started:	Wed, Feb 1, 2023, 7:26 PM
Interval:	10ms
Buffer capacity:	1GB
Buffer duration:	Unlimited
Features:	js, screenshots, stackwalk, nostacksampling, ipcmessages, cpu, markersallthreads, processcpu, power

Examples

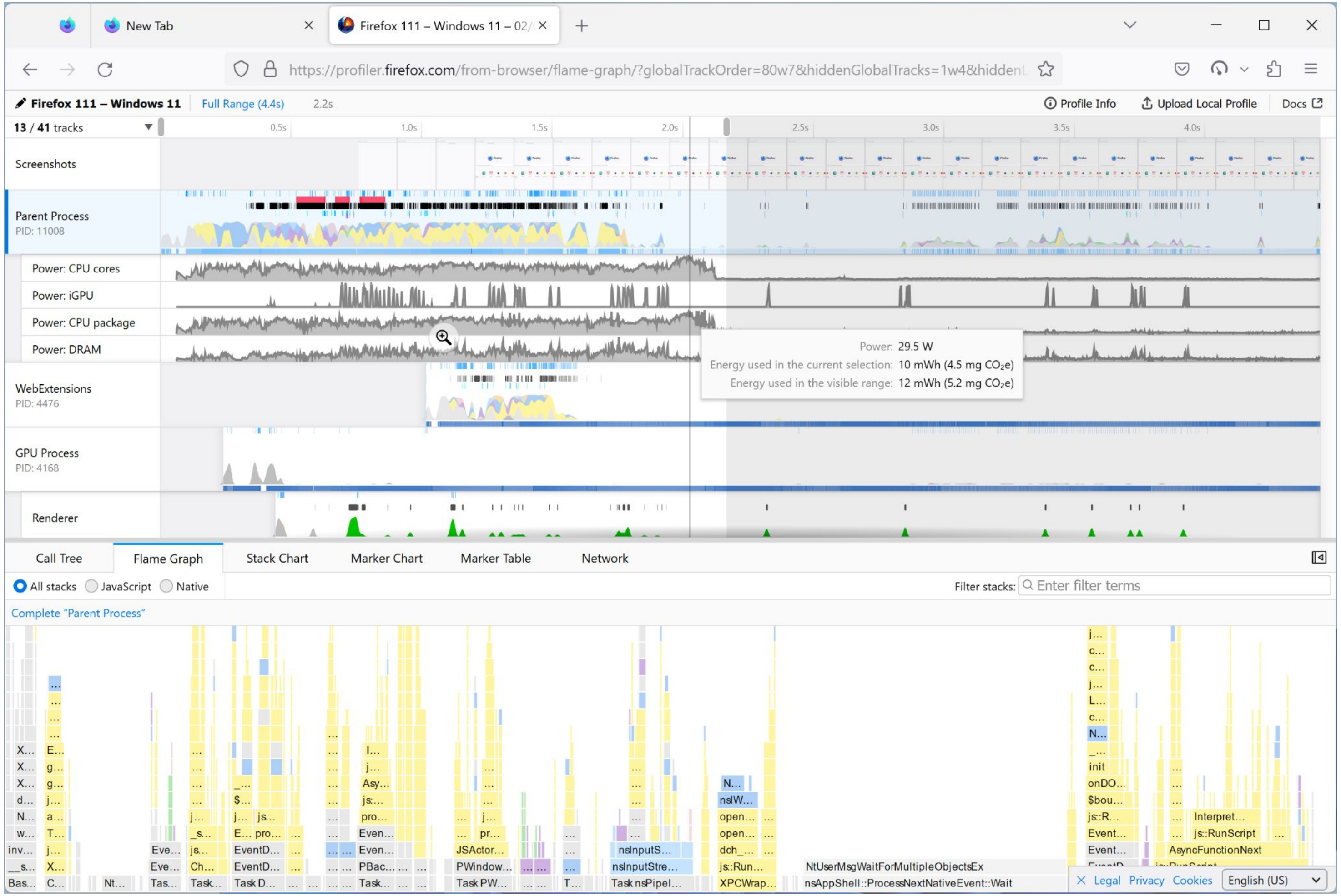
Link to slides:
https://share.firefox.dev/power-profiling-fosdem2024



```
Power: 5.73 W
Energy used in the current selection: 134 µWh (0.059 mg CO₂e)
Energy used in the visible range: 135 µWh (0.060 mg CO₂e)
```

```
http://en.wikipedia.org/
https://en.wikipedia.org/
https://en.wikipedia.org/wiki/Main_Page
https://en.wikipedia.org/w/load.php?lang=en&mod...
https://en.wikipedia.org/w/load.php?lang=en&mod...
https://en.wikipedia.org/w/load.php?lang=en&mod...
https://en.wikipedia.org/static/images/i... /wikipedia.png
https://en.wikipedia.org/static/images/... /wikipedia-wordmark-en.svg
https://en.wikipedia.org/static/images/... /wikipedia-tagline-en.svg
https://upload.wikimedia.org/wikipedia/com... /376px-Resolution_Guyout_3D_relief_%28transparent%29.png
https://upload.wikimedia.org/wikipedia/com... /240px-Kanji_kato.jpg
https://upload.wikimedia.org/wikipedia/com... /342px-C2022_E3_%28ZTF%29-_Alessandro_Bianconi.png
https://upload.wikimedia.org/wikipedia/com... /324px-Stellenbosch-region.JPG
https://upload.wikimedia.org/wikipedia/com... /600px-Wesley_Bell_2014_-_edit2.jpg
https://upload.wikimedia.org/wikipedia/en/t... /62px-Commons-logo.svg.png
https://upload.wikimedia.org/wikipedia/com... /70px-MediaWiki-2020-icon.svg.png
```

Loading Wikipedia homepage - https://share.firefox.dev/3RqH4Ke

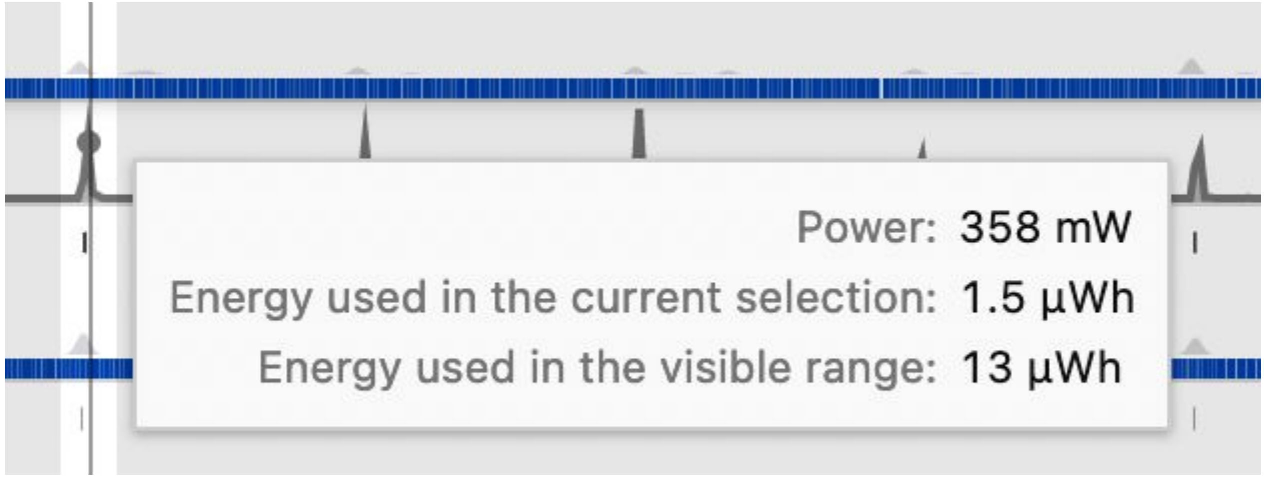


Power: 29.5 W
Energy used in the current selection: 10 mWh (4.5 mg CO₂e)
Energy used in the visible range: 12 mWh (5.2 mg CO₂e)

Starting Firefox - https://share.firefox.dev/3X0PHMP

Measure tiny things

What’s the smallest thing we can power profile?



Power: 358 mW

Energy used in the current selection: 1.5 µWh

Energy used in the visible range: 13 µWh

Showing about:blank, cursor in the address bar - https://share.firefox.dev/3U8hLgp

Demo

Zooming on openstreetmap - https://share.firefox.dev/3UjDIfy

Demo

Recap

Demo

Android?

External power profiling

AC power measure

CPU power data

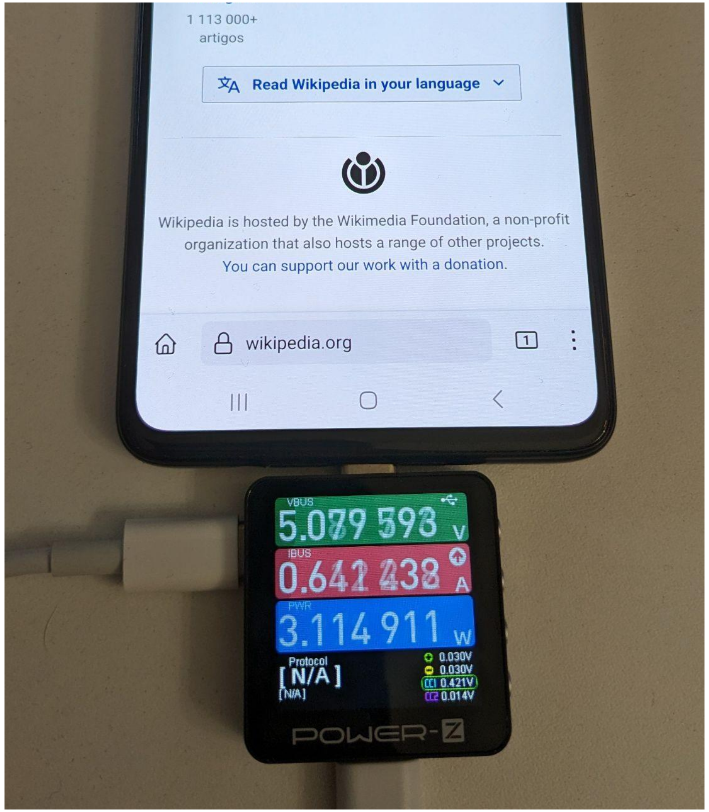
?

Babylonfive David W. Smith, CC BY 3.0, via Wikimedia Commons

Max 50Hz sampling

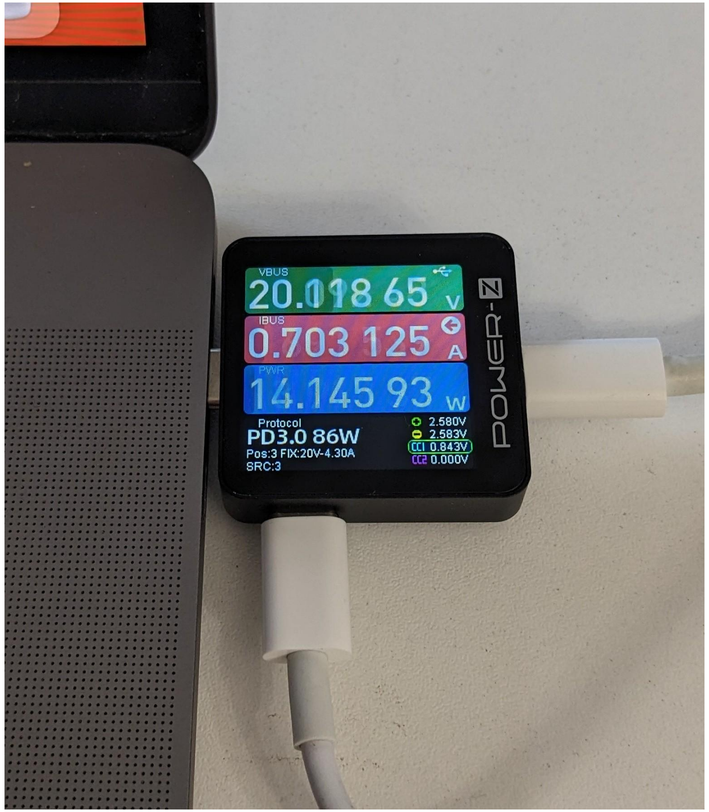
Not profiling the entire computer

Charger testers


1 113 000+
artigos
Read Wikipedia in your language
Wikipedia is hosted by the Wikimedia Foundation, a non-profit organization that also hosts a range of other projects.
You can support our work with a donation.
wikipedia.org
VBUS
5.079 593 V
IBUS
0.641 438 A
PWR
3.114 911 W
Protocol
[N/A]
[N/A]
0.030V
0.030V
CC1 0.421V
CC2 0.014V
POWER-Z

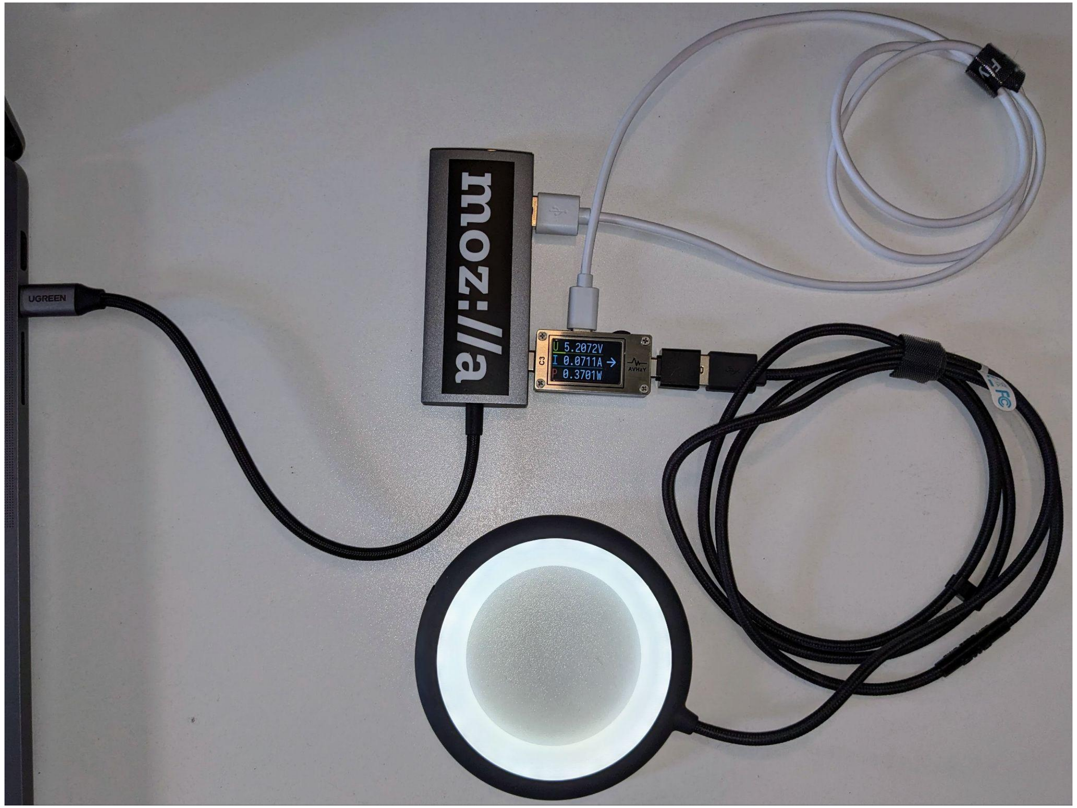
- Affordable (<100€) devices made to verify how good chargers are.
- Up to 1kHz sampling
- Some can export data to a computer through USB or Bluetooth.
- When the battery is full, they measure how much power is used by the phone.

Charger testers also work for laptops



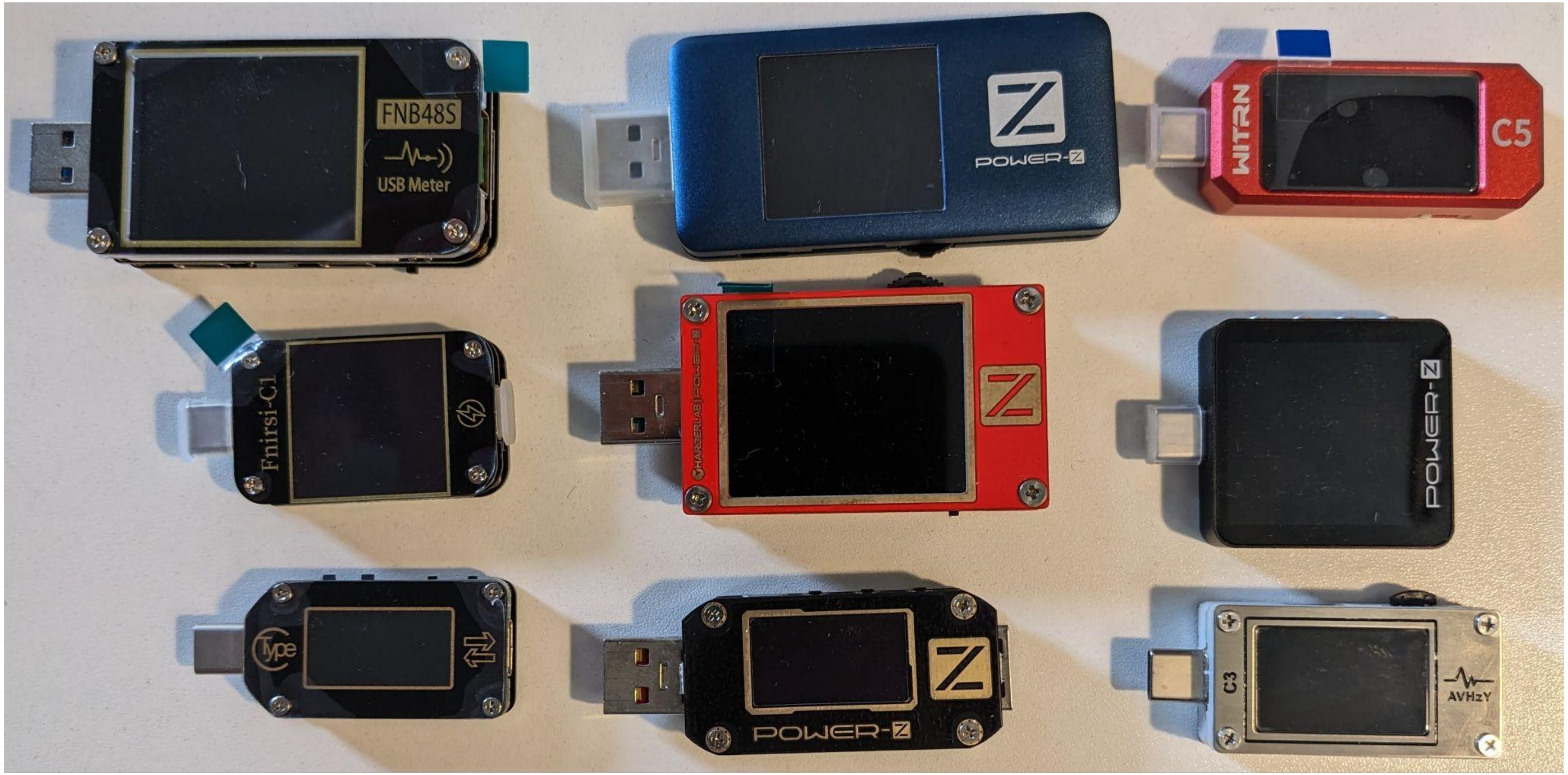
- Supports up to 240W
- Modern laptops support charging through USB PD (Power Delivery)
- Support for “external power profiling” landed in Firefox 121.

Testing and reverse engineering



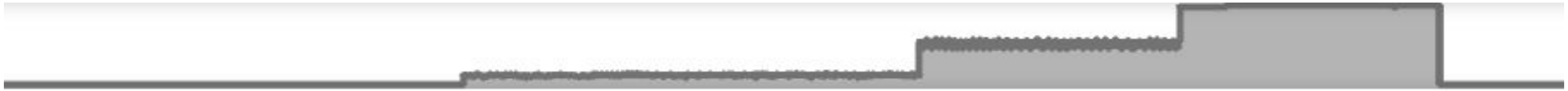
- Software only for Windows
- Mix of Chinese and English
- Poorly documented APIs (if documented at all)
- Tested with a USB light

Compatible USB power meters

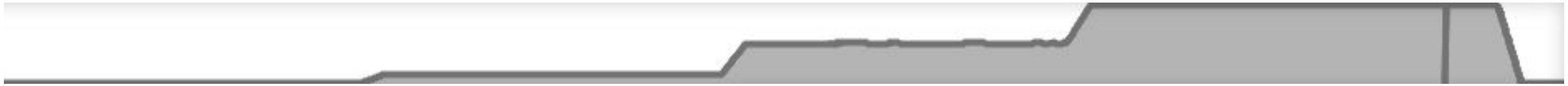


Plug & play!

- The models on the picture “just work”
- https://github.com/fqueze/usb-power-profiling
- Example profiles for each supported device:
  - Good example:

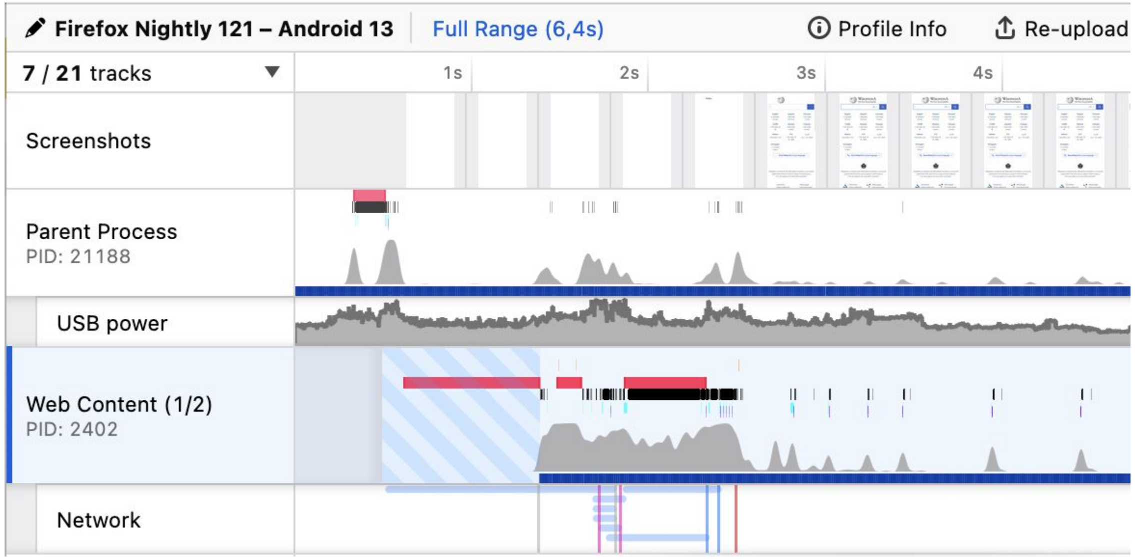


  - Bad example:



- Data source for the “USB power” track in examples for the next slides.

Android remote + ext. power



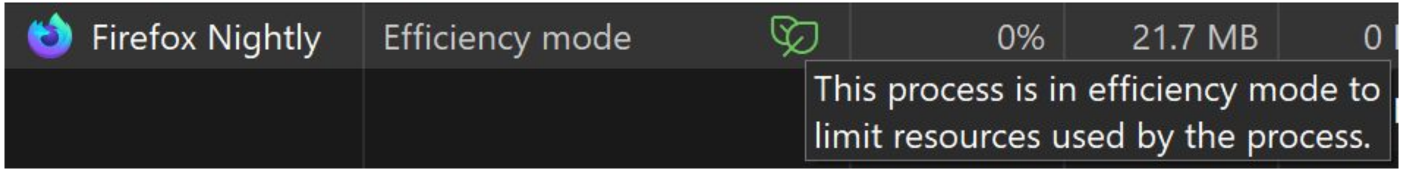
The power used by the idle phone is around 2W. Goes up to 7W during page load.

Android wikipedia page load - https://share.firefox.dev/3QTmhii

More examples

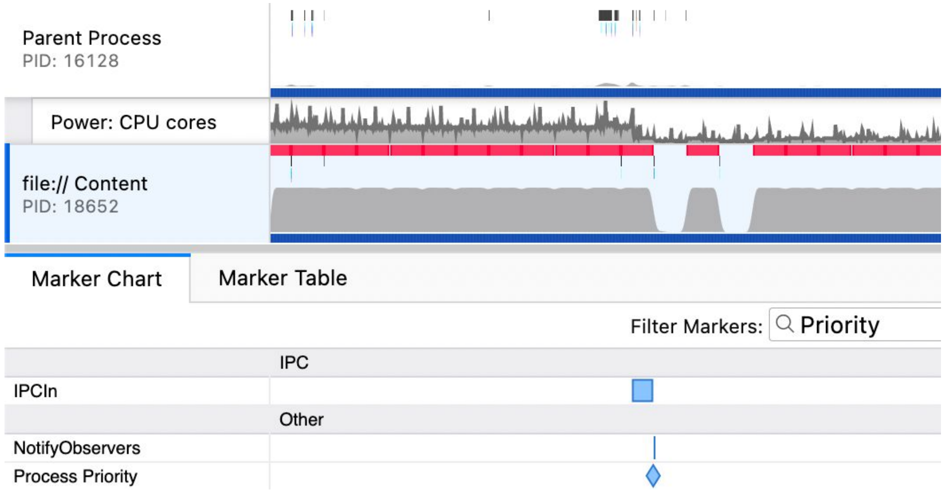
Link to slides:
https://share.firefox.dev/power-profiling-fosdem2024

Using efficiency mode on Windows 11



Firefox Nightly | Efficiency mode | 0% | 21.7 MB | 0

This process is in efficiency mode to limit resources used by the process.

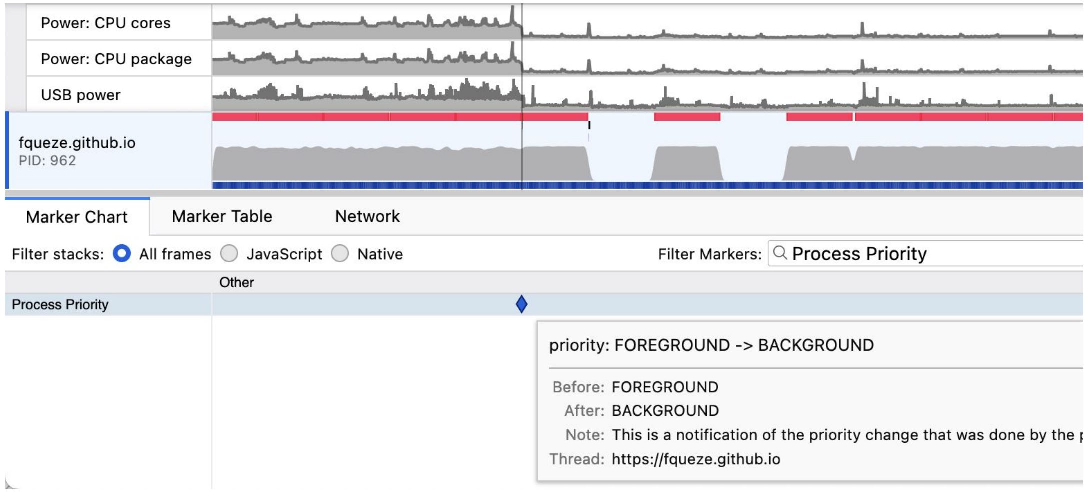


The power used by using 100% of a core drops from 10W to 2W.

Win11 "Efficiency mode" (EcoQos) - https://share.firefox.dev/3FJoAkc (bug 1796525 Firefox 108)

Using 'background' QoS on Mac

1 s of CPU time



priority: FOREGROUND -> BACKGROUND

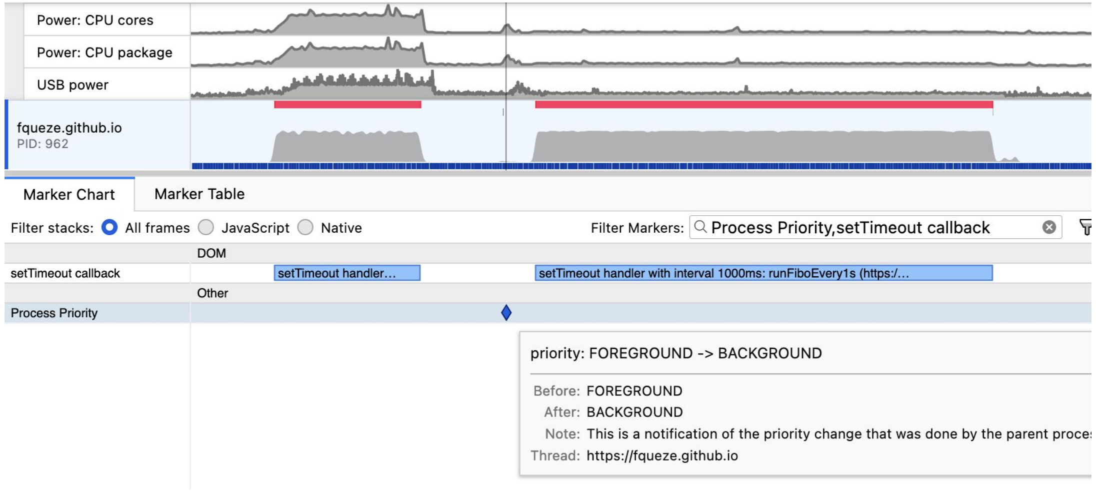
Before: FOREGROUND
After: BACKGROUND
Note: This is a notification of the priority change that was done by the p
Thread: https://fqueze.github.io

The power used by cores drops from 18W to 1.6W.
Entire MacBook from 30W to 10W.

Mac 'background' QoS - https://share.firefox.dev/3QRSyX1 (Bug 1805932 - Nightly 115+)

Using ‘background’ QoS on Mac

Running the same task


Power: CPU cores
Power: CPU package
USB power
fqueze.github.io
PID: 962
Marker Chart
Marker Table
Filter stacks: All frames JavaScript Native
Filter Markers: Process Priority,setTimeout callback
DOM
setTimeout callback
setTimeout handler…
setTimeout handler with interval 1000ms: runFiboEvery1s (https:/…
Other
Process Priority

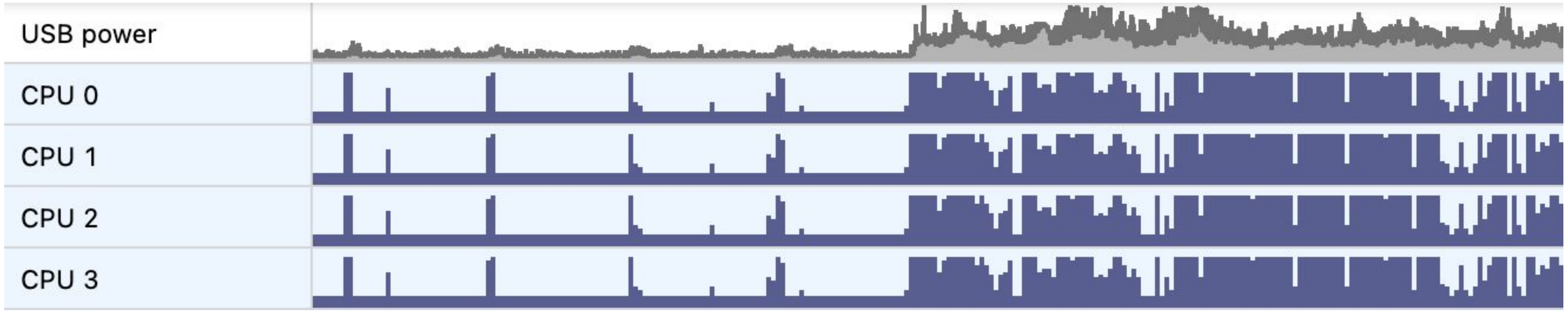
priority: FOREGROUND -> BACKGROUND

Before: FOREGROUND
After: BACKGROUND
Note: This is a notification of the priority change that was done by the parent proce
Thread: https://fqueze.github.io

	Foreground	Background
Time	500ms	1.6s
CPU energy	3mWh	1.2mWh
Laptop energy	4.1mWh	5mWh

Mac ‘background’ QoS - https://share.firefox.dev/3STHhsa

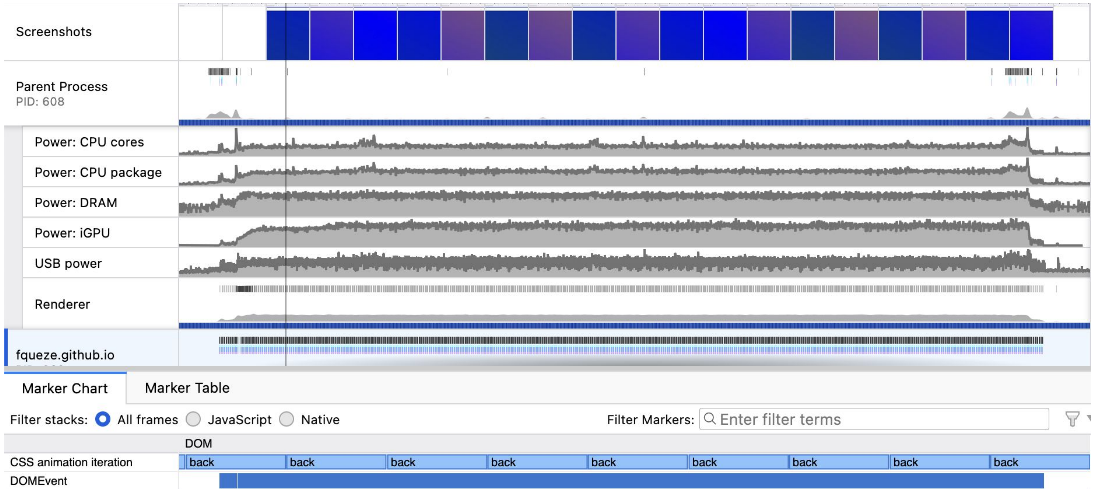
Raising the CPU clock frequency



Power and CPU frequency profiling on Android - https://share.firefox.dev/3Uqh45j

Animated background

Burning your lap, millions of pixels at a time



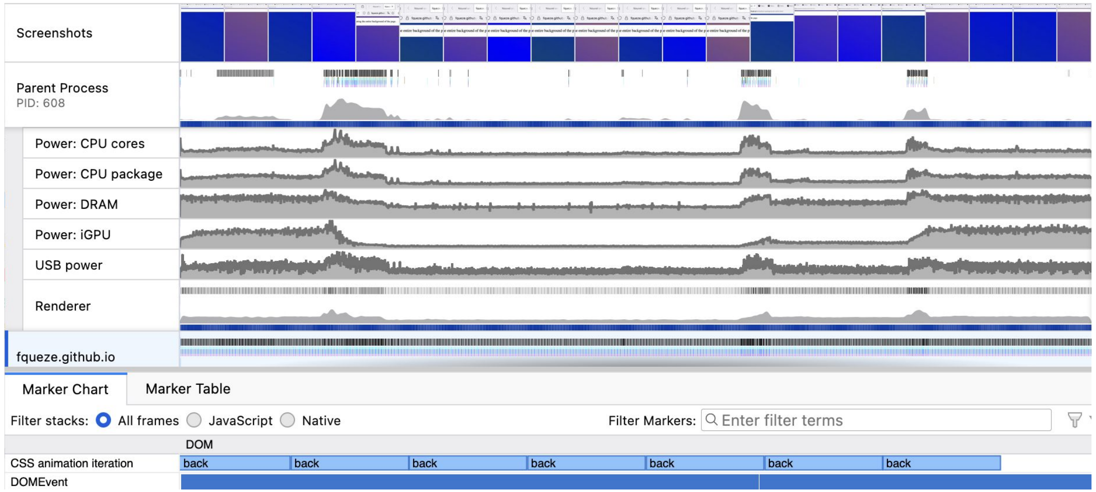
The GPU power is as high as the CPU core power.

DRAM power is significant too.

Animated background - https://share.firefox.dev/3GbKA6o

Animated background

In different window sizes



GPU power:

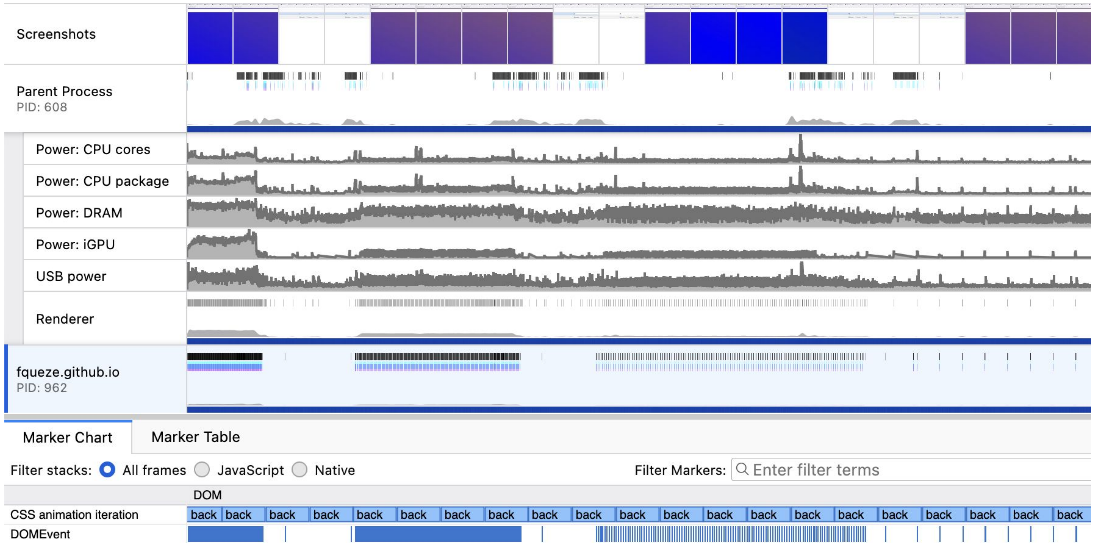
- 0.3W for 0.45Mpixels
- 1.3W for 2.4Mpixels
- 6W for 6.3Mpixels

CPU power dominates while resizing

Animated background at different sizes - https://share.firefox.dev/3uB0sws

Animated background

At different refresh rates



Average GPU power:

- 4.5W for 60Hz
- 1.3W for 30Hz
- 0.4W for 10Hz
- 0.05W for 1Hz

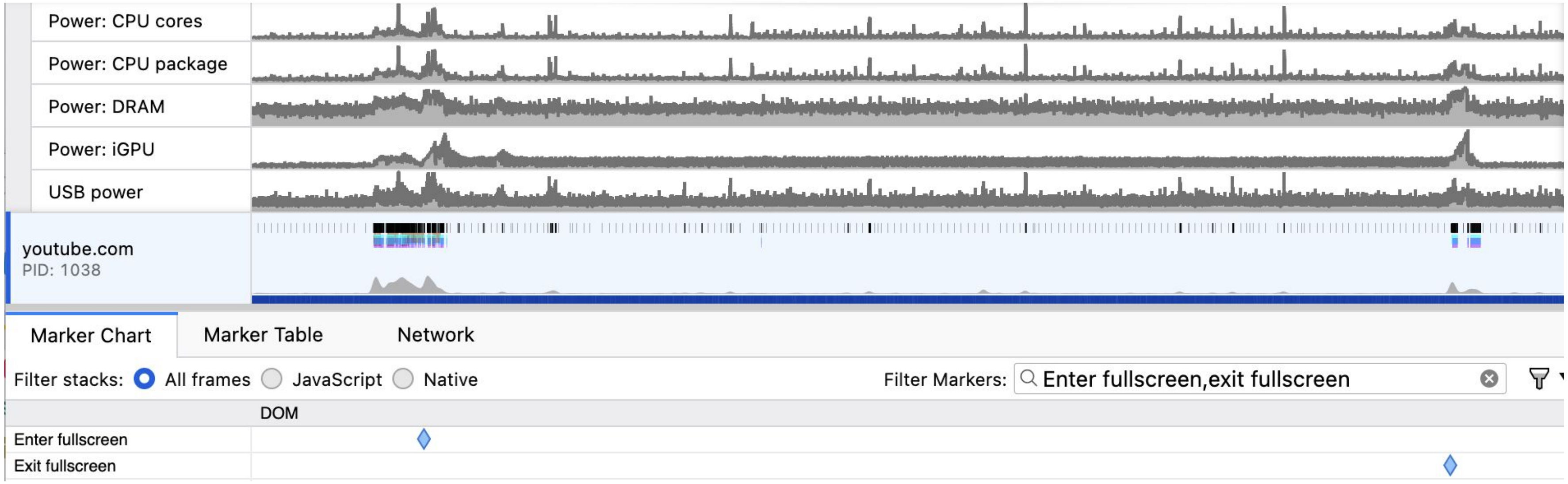
<30Hz, a spike per frame

Animated background at different fps - https://share.firefox.dev/3RbY1t6

Playing a video

In a frame or fullscreen

CPU and GPU power use comparable to the animated background.

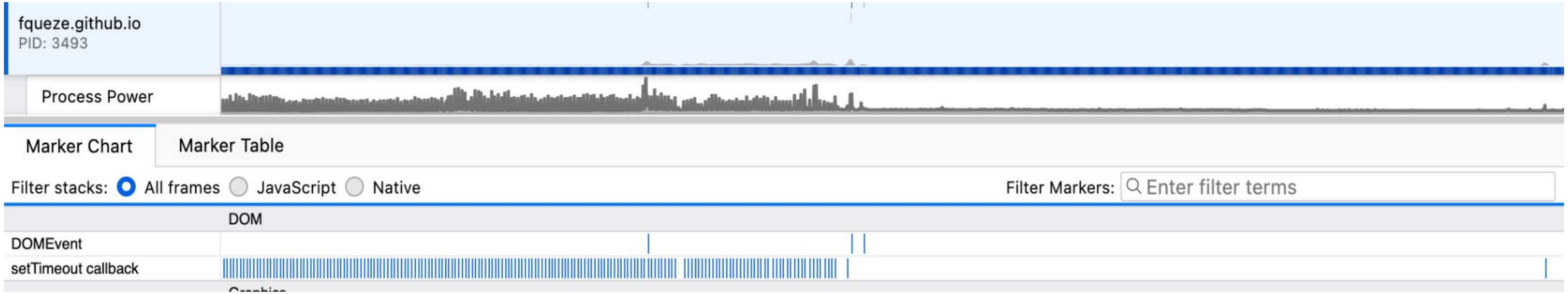


Note: 30fps on the video.

Video playback - https://share.firefox.dev/3R9EzgD

Waking up... too often

setTimeout(0)



First half: tab is visible, timer wake-ups every 4ms.

Second half: tab in the background, timers throttled to 1Hz.

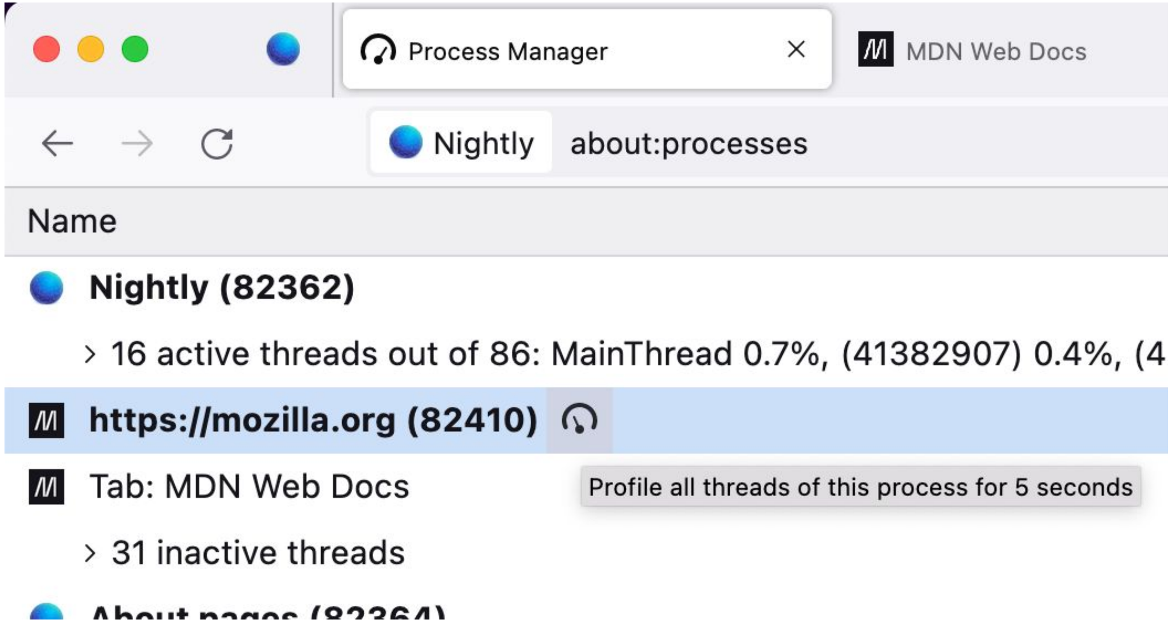
Timers - https://share.firefox.dev/499xUcB

Bonus

A few more things

Firefox task manager

With one click profiling (the 'power' feature is enabled)



Name

Nightly (82362)

› 16 active threads out of 86: MainThread 0.7%, (41382907) 0.4%, (4

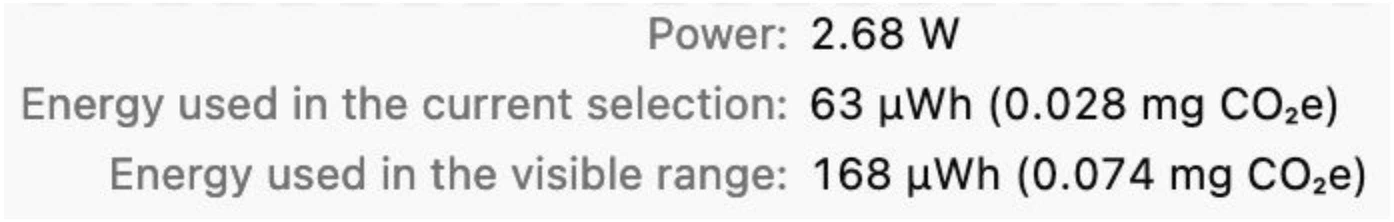
https://mozilla.org (82410)

Tab: MDN Web Docs

Profile all threads of this process for 5 seconds

› 31 inactive threads

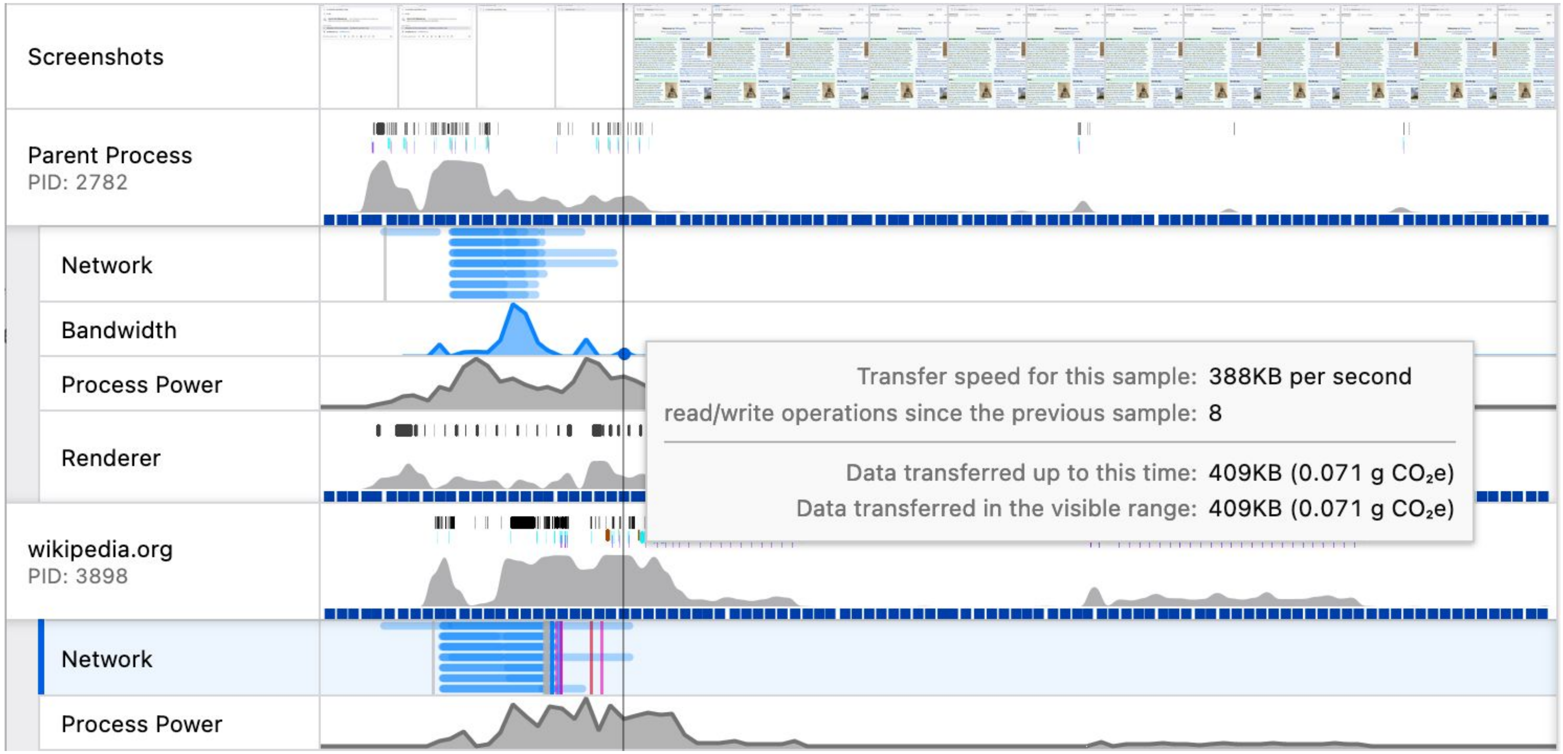
CO_2 equivalent using co2.js



Power:	2.68 W
Energy used in the current selection:	63 µWh (0.028 mg CO_2e)
Energy used in the visible range:	168 µWh (0.074 mg CO_2e)

Thanks to **Chris Adams** and **Fershad** from **The Green Web Foundation**.

Network bandwidth

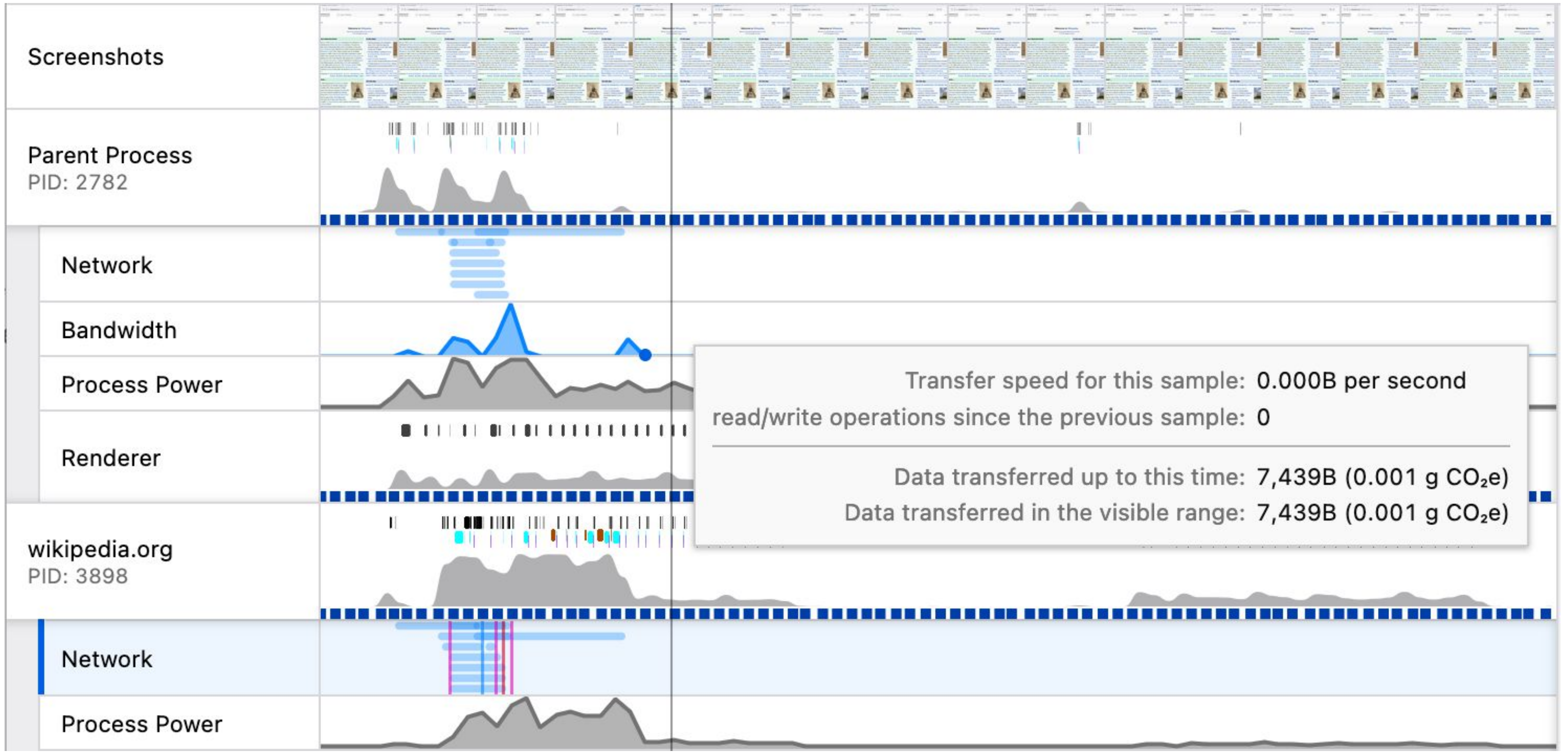


Transfer speed for this sample: 388KB per second
read/write operations since the previous sample: 8

Data transferred up to this time: 409KB (0.071 g CO₂e)
Data transferred in the visible range: 409KB (0.071 g CO₂e)

Loading the wikipedia home page twice - https://share.firefox.dev/3SfHljL (Bandwidth track in Firefox 123)

Network bandwidth (cached)

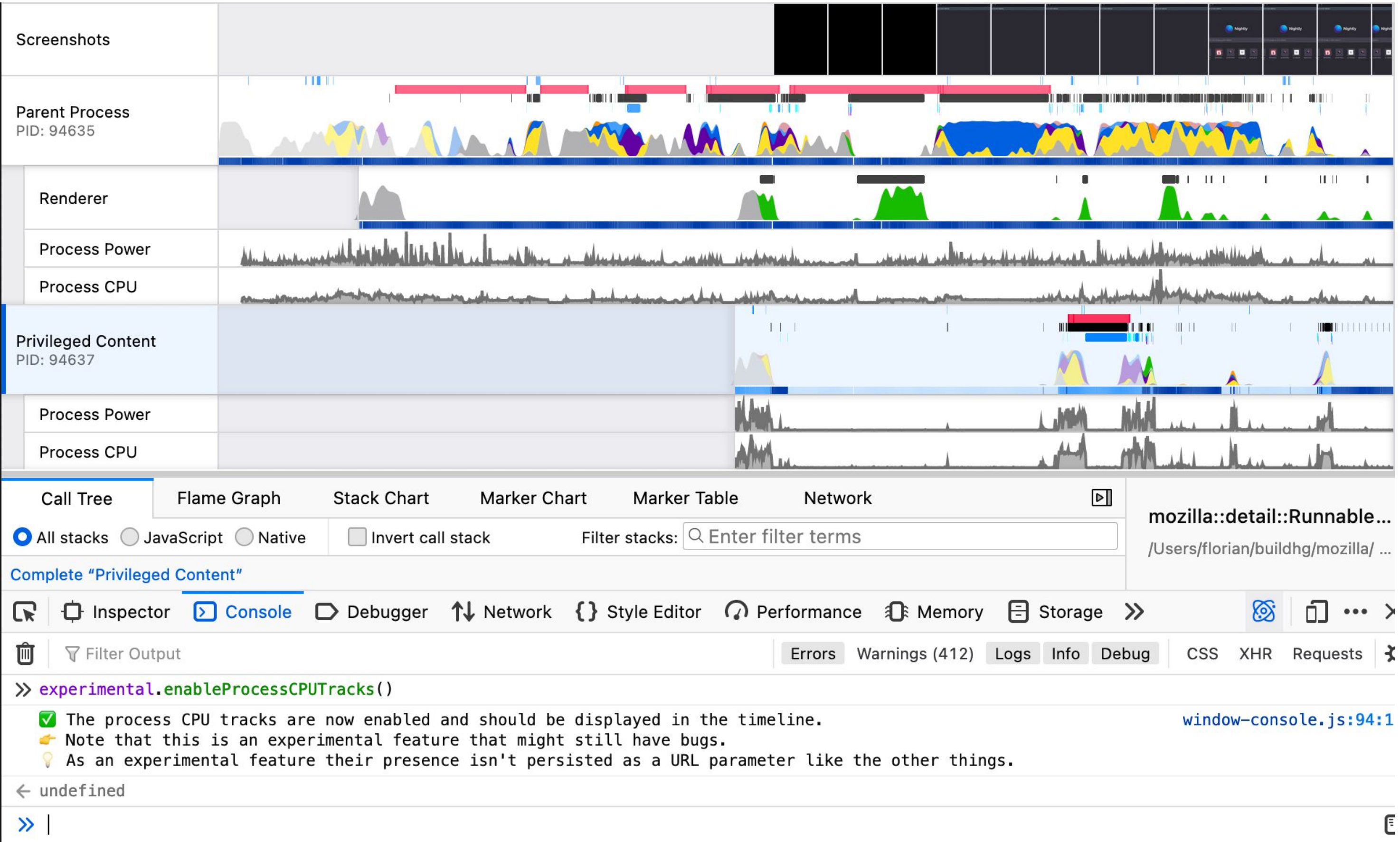


Transfer speed for this sample: 0.000B per second
read/write operations since the previous sample: 0

Data transferred up to this time: 7,439B (0.001 g CO₂e)
Data transferred in the visible range: 7,439B (0.001 g CO₂e)

Loading the wikipedia home page twice - https://share.firefox.dev/3SfHIjL (Bandwidth track in Firefox 123)

Process CPU



```
>> experimental.enableProcessCPUTracks()
✅ The process CPU tracks are now enabled and should be displayed in the timeline.      window-console.js:94:1
👉 Note that this is an experimental feature that might still have bugs.
💡 As an experimental feature their presence isn't persisted as a URL parameter like the other things.
← undefined
>> |
```

When power profiling is not supported on your machine...

Conclusion

- **Power profiling is:**
  - Possible
  - Easy
  - Fun
- **But start with**
  - Web compat
  - Good performance

Thanks! Questions?

- Share ideas, #profiler:mozilla.org
- Questions: florian@mozilla.com

bitly

Link to slides:
https://share.firefox.dev/power-profiling-fosdem2024