



Power profiling with the Firefox Profiler

Florian Quèze

February 4, 2023 - FOSDEM

Table of contents

- **What's the Firefox Profiler?**
- **Power profiling**
 - Why?
 - How it happened
 - Supported platforms
- **Examples**
- **Bonus**

The logo for Firefox Profiler is a large, dark blue circle on the left side of the page. It is partially overlapped by a bright pink shape at the top and a red shape at the bottom. The text "Firefox Profiler" is written in white, bold, sans-serif font inside the dark blue circle.

Firefox Profiler

<https://profiler.firefox.com>

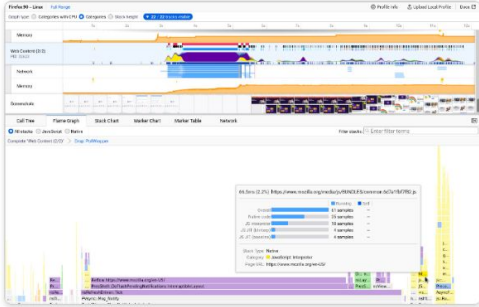
What's the Firefox Profiler

- Built-into Firefox
- Initially for performance work
- Main sources of data:
 - Samples
 - Stacks
 - Counters
 - Markers



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!

Legal Privacy Cookies English (US)



Firefox Profiler — Web app for Firefox performance analysis

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

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

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

Legal Privacy Cookies English (US)



Firefox Profiler — Web app for Firefox performance analysis

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

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

Recording...

Discard **Capture**

^⌘1 ^⌘2

To start profiling, click on the profiler icon in the browser toolbar. The icon is located in the top right corner of the browser window. 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.](#)

[Legal](#) [Privacy](#) [Cookies](#) English (US) ▾



Firefox 111 - macOS 12.3.1 - 2/1 ×

https://profiler.firefox.com/from-browser/calltree?globalTrackOrder=g0wf&hiddenGlobalTracks=1we&hiddenL

Full Range (3.4s) > 875ms

Firefox 111 - macOS 12.3.1

7 / 42 tracks

Screenshots

Parent Process
PID: 84177

Network

Memory

wikipedia.org (2/2)
PID: 91497

Network

Memory

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::M...
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::M...
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::M...
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
JSRuntime(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

```

Call Tree

Flame Graph

Stack Chart

Marker Chart

Marker Table

All stacks JavaScript Native Invert call stack

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

Total (samples)	Self	
100%	869	—
100%	869	—
100%	869	—
100%	869	—
100%	869	—
100%	869	—
100%	869	—
100%	869	—
100%	869	—
100%	869	—
100%	869	—
100%	869	—
100%	869	—
100%	869	—
100%	869	—
100%	869	—
100%	869	—

Running sample count 100% 1

Self sample count 100% 1



Wikipedia, the free encyclopedia X Firefox 111 - macOS 12.3.1 - 2/1 X

← → ↻ https://profiler.firefox.com/from-browser/calltree?globalTrackOrder=g0wf&hiddenGlobalTracks=1we&hiddenL ☆

Firefox 111 - macOS 12.3.1 Full Range (3.4s) > 875ms Profile Info Upload Local Profile Docs

7 / 42 tracks 1.45s 1.50s 1.55s 1.60s 1.65s 1.70s 1.75s 1.80s 1.85s 1.90s 1.95s 2.00s 2.05s 2.10s 2.15s 2.20s 2.25s

Screenshots

Parent Process PID: 84177

Network

Memory

wikipedia.org (2/2) PID: 91497

Network

Memory

Call Tree Flame Graph Stack Chart Marker Chart Marker Table

All stacks JavaScript Native Invert call stack

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

filter terms

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	—	MessageLoop::RunHandler() /Users/florian/buildhg/mozilla/ipc/chromium/src/base/message_loop.cc
100%	869	—	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	—	MessageLoop::RunHandler() /Users/florian/buildhg/mozilla/ipc/chromium/src/base/message_loop.cc
100%	869	—	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/MessagePump.cpp
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

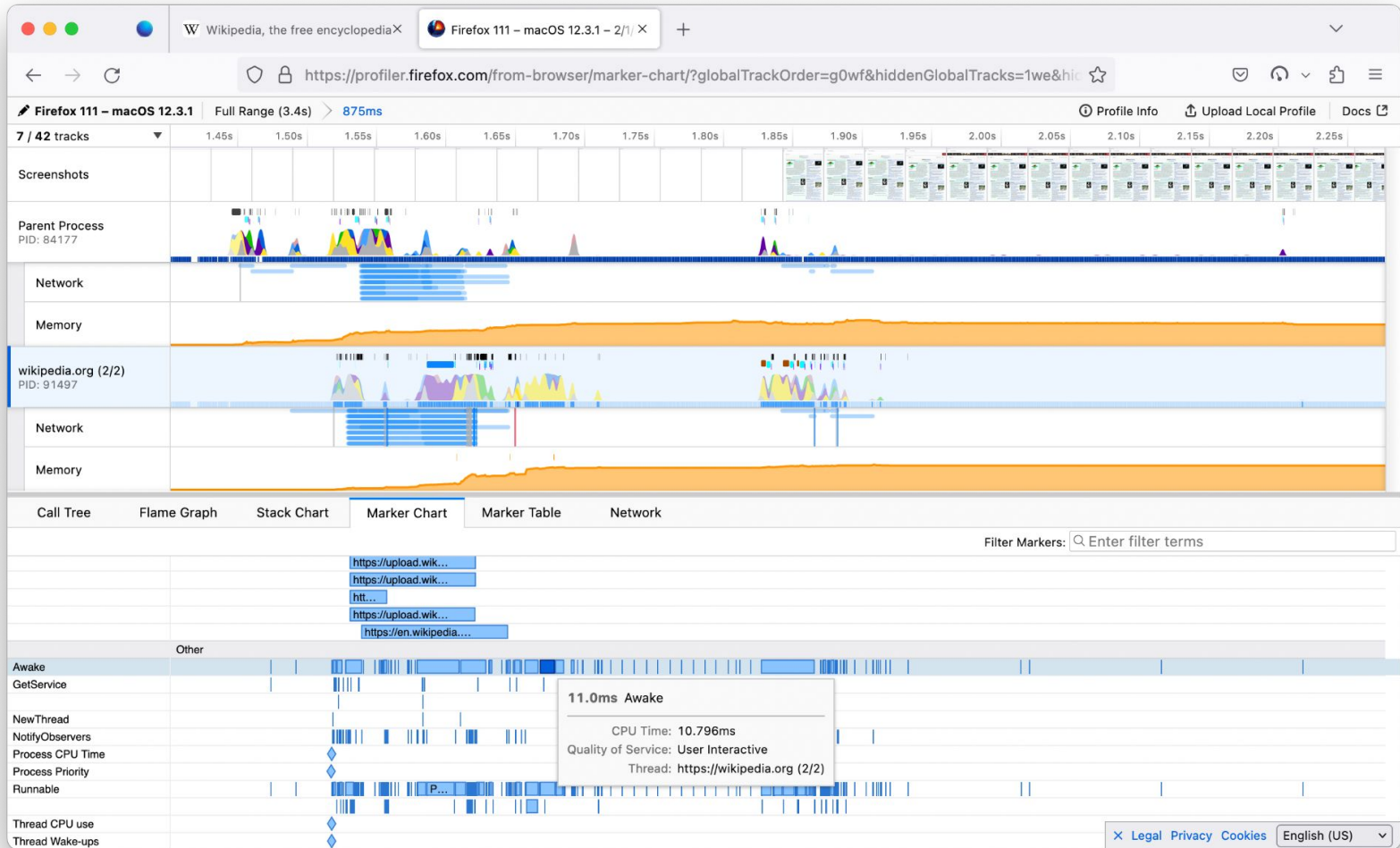
JavaScript 100% 1

Categories Self sample count

JavaScript 100% 1

Legal Privacy Cookies English (US)







Power profiling

Why? User experience!

Excessive power use = poor user experience

- Noisy fans
- Hot laptops
- Short battery life



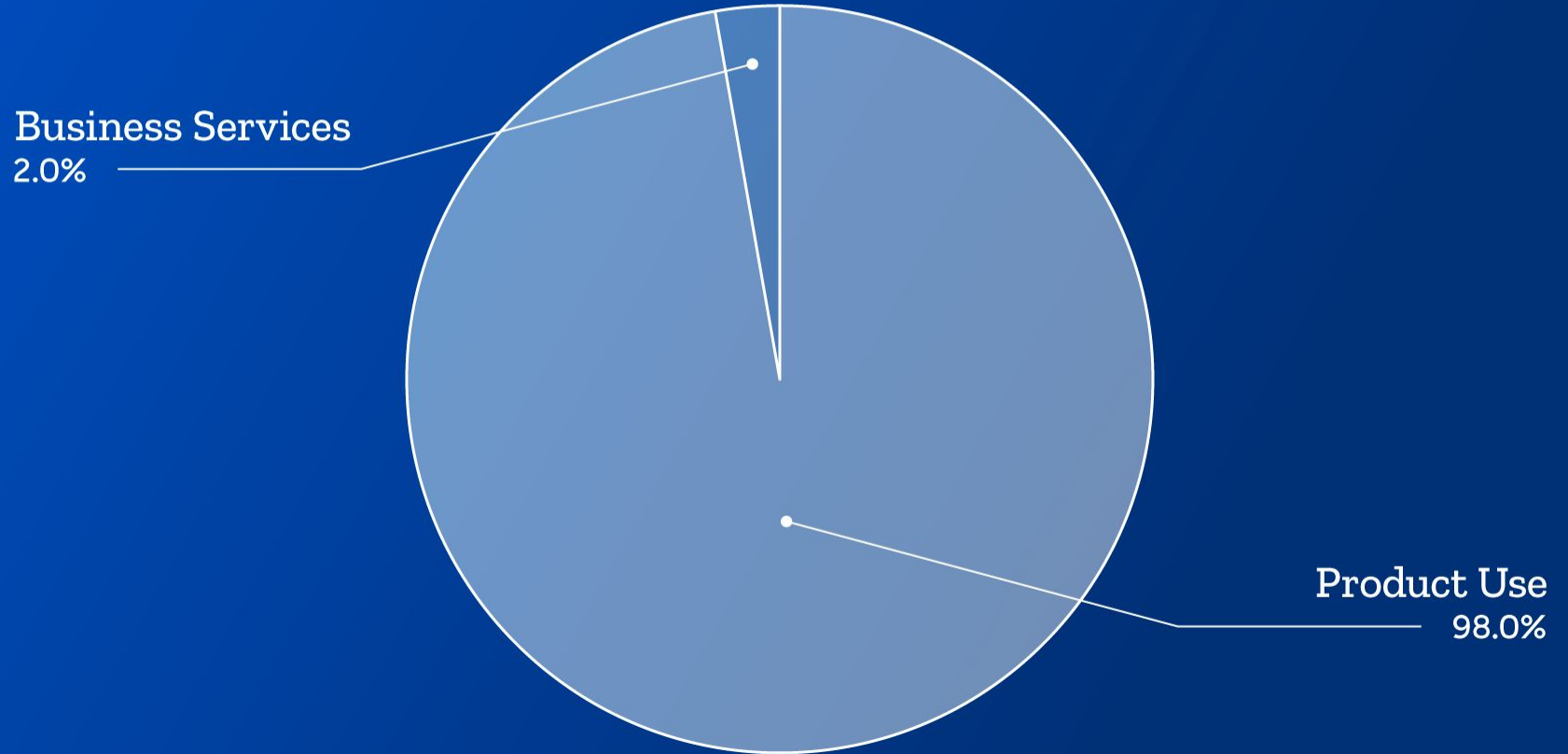
Why? Sustainability!

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 2019





Understanding local power use

(cheap) Wattmeter



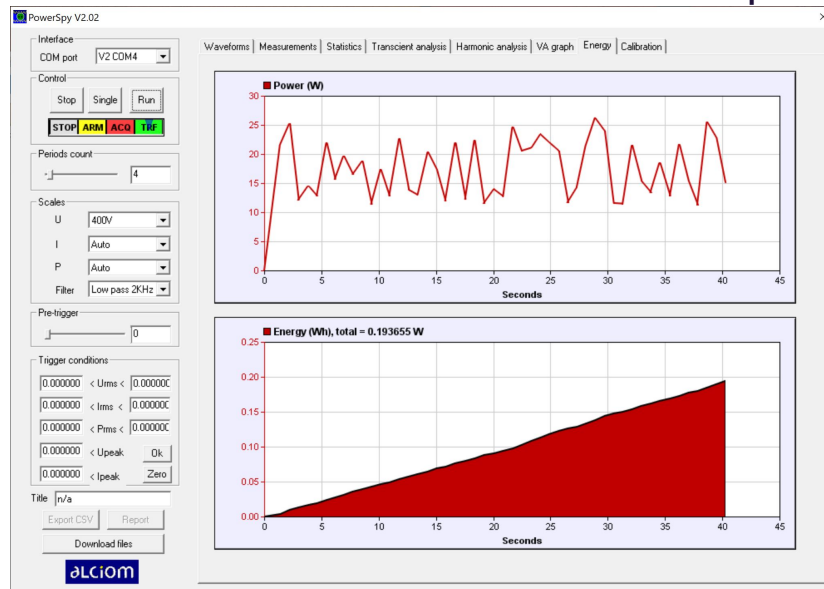
- Affordable
- Reasonably accurate
- Can't track things over time



Better wattmeter



- Connects over bluetooth
- Data readable from another computer



- Still difficult to match with what was done



How Microsoft did 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.

...

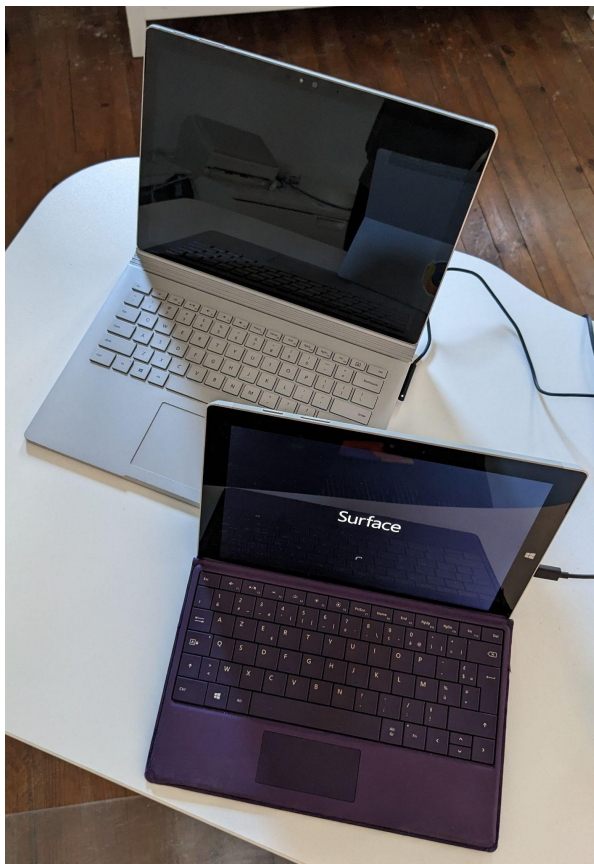
Measuring power

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:

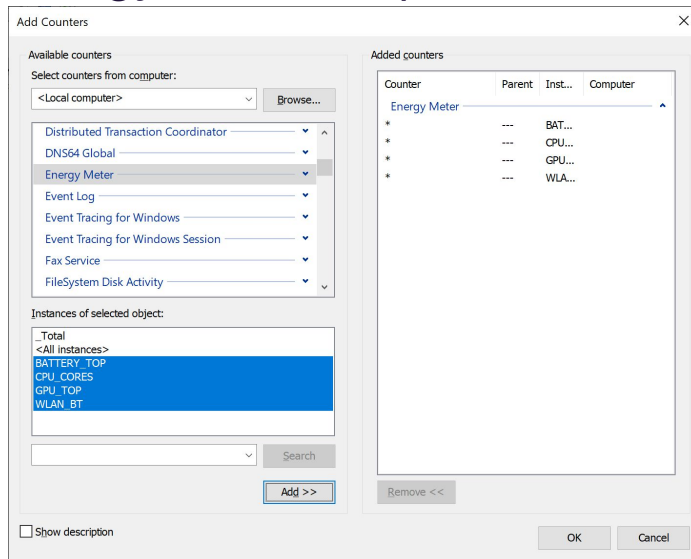
- \Energy Meter (CPU_CORES)\Power
- \Energy Meter (GPU_TOP)\Power
- \Energy Meter (WLAN_BT)\Power



Devices with MAXIM power meters

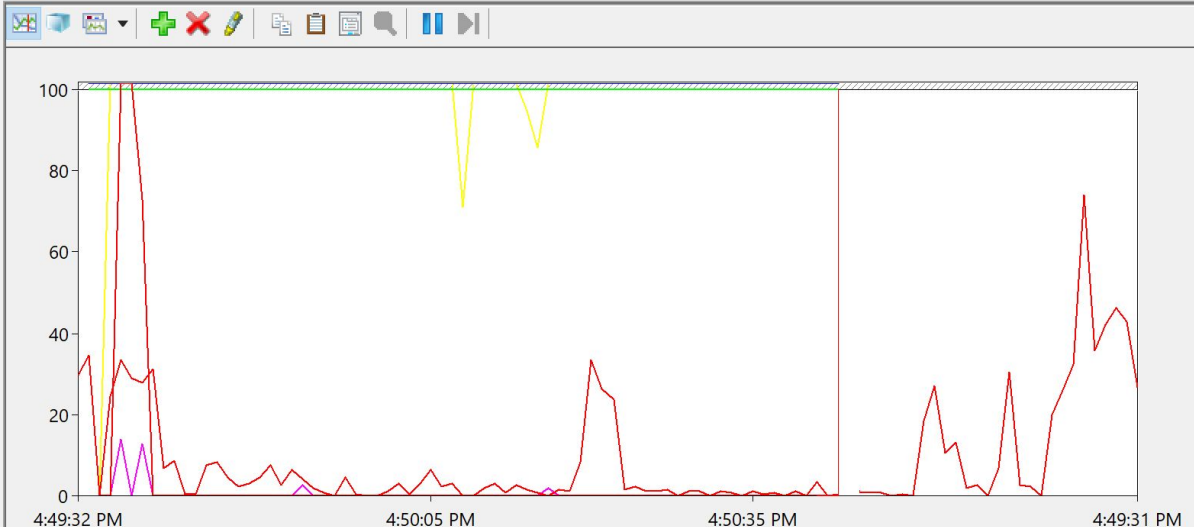


- Surface Book 1, Surface 3
- Many other devices expose it in their ACPI table, but don't actually have the chip
- Energy meters in perfmon.exe:





- Performance
 - Monitoring Tools
 - Performance Monitor
 - Data Collector Sets
 - Reports

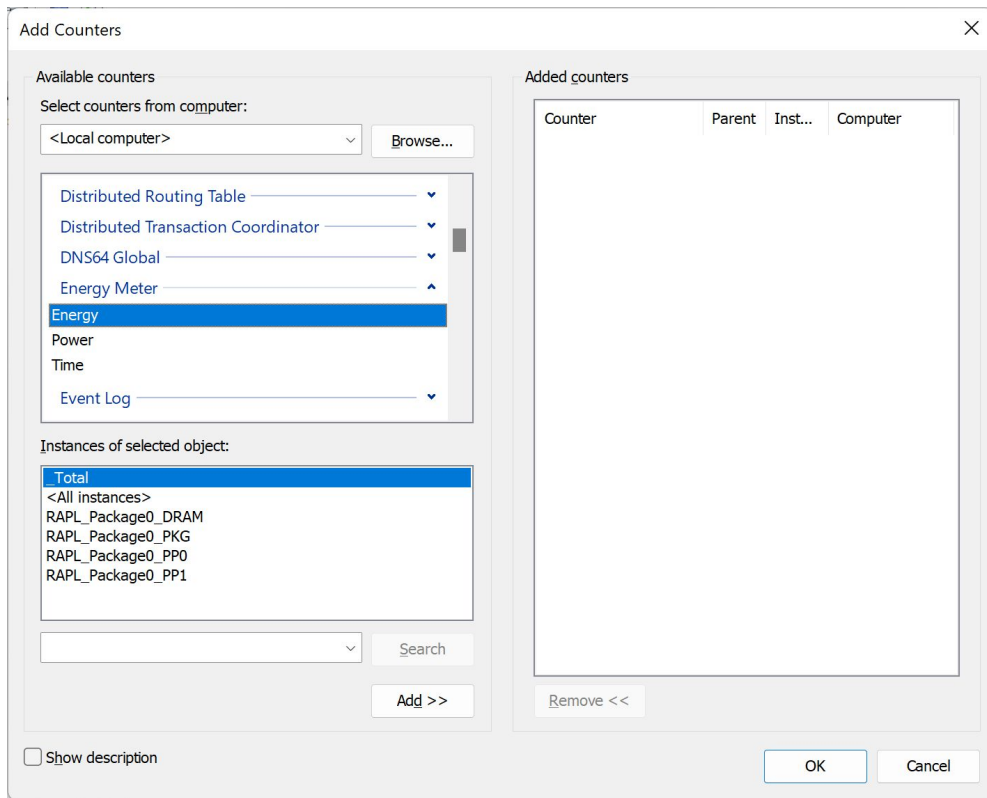


Last: 5.0539e+011 Average: 4.9860e+011 Minimum: 4.8417e+011 Maximum: 5.0539e+011
 Duration: 1:40

Show	Color	Scale	Counter	Instance	Parent	Object	Computer
<input checked="" type="checkbox"/>	—	1.0	% Processor Time	_Total	---	Processor Information	\\DESKTOP-BIB352E
<input checked="" type="checkbox"/>	—	1.0	Power	BATTERY_T...	---	Energy Meter	\\DESKTOP-BIB352E
<input checked="" type="checkbox"/>	—	1.0	Energy	BATTERY_T...	---	Energy Meter	\\DESKTOP-BIB352E
<input checked="" type="checkbox"/>	—	1.0	Time	BATTERY_T...	---	Energy Meter	\\DESKTOP-BIB352E
<input checked="" type="checkbox"/>	—	1.0	Power	CPU_CORES	---	Energy Meter	\\DESKTOP-BIB352E
<input checked="" type="checkbox"/>	—	1.0	Energy	CPU_CORES	---	Energy Meter	\\DESKTOP-BIB352E
<input checked="" type="checkbox"/>	—	1.0	Time	CPU_CORES	---	Energy Meter	\\DESKTOP-BIB352E
<input checked="" type="checkbox"/>	—	1.0	Power	GPU_TOP	---	Energy Meter	\\DESKTOP-BIB352E



A good surprise...



- 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 X

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

Screenshots

Parent Process
PID: 9544

Network

Timer

RAPL_Package0_PKG

RAPL_Package0_PP0

RAPL_Package0_PP1

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

All stacks JavaScript Native Invert call stack

Filter stacks:

Complete "Parent Process"

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

Select a node to display information about it.

Legal Privacy Cookies English (US)



Power profiling - platform support

Windows:

- 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 (TBC) - AMD Ryzen CPUs
1 track per core!



Power profiling - platform support

Mac:

- 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 [8 years old implementation](#))



Power profiling - platform support

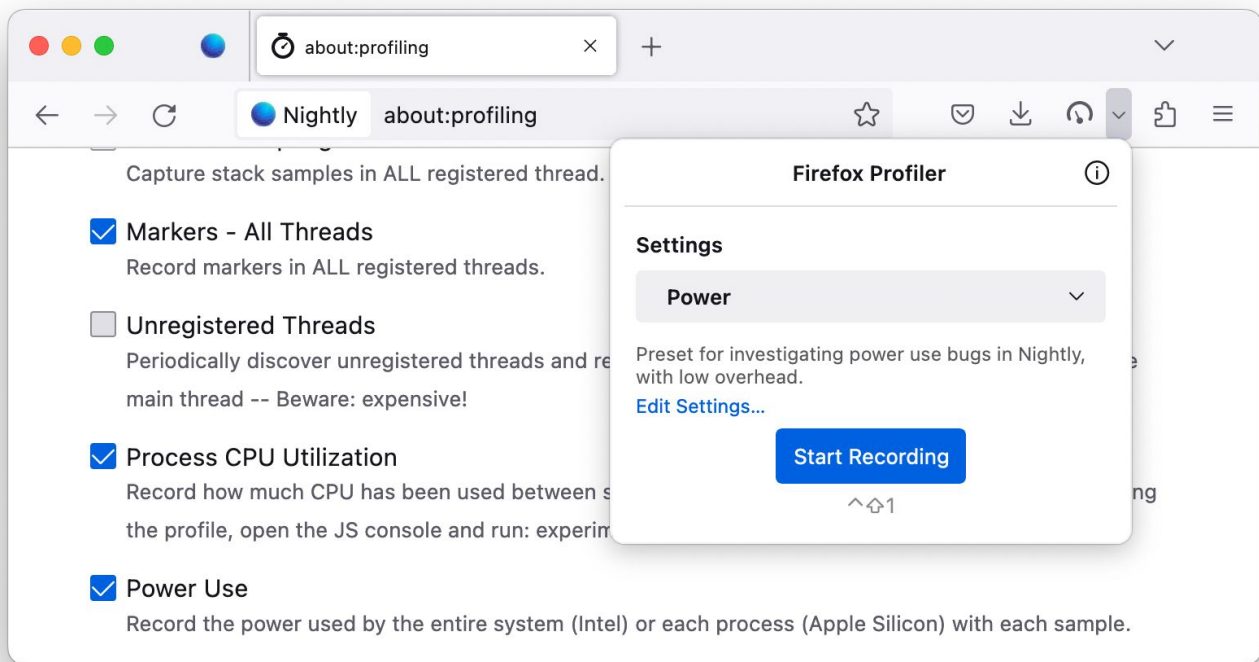
Linux:

- [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



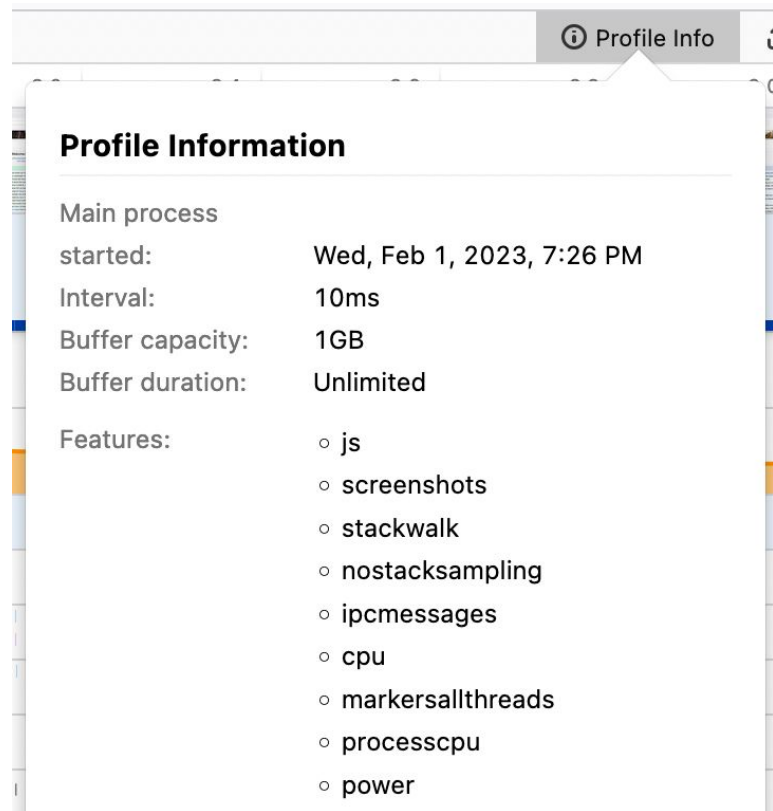
Power profiling - configuration

There is a 'Power' preset for easy configuration.



Reducing overhead

- Longer interval
- No periodic stack sampling



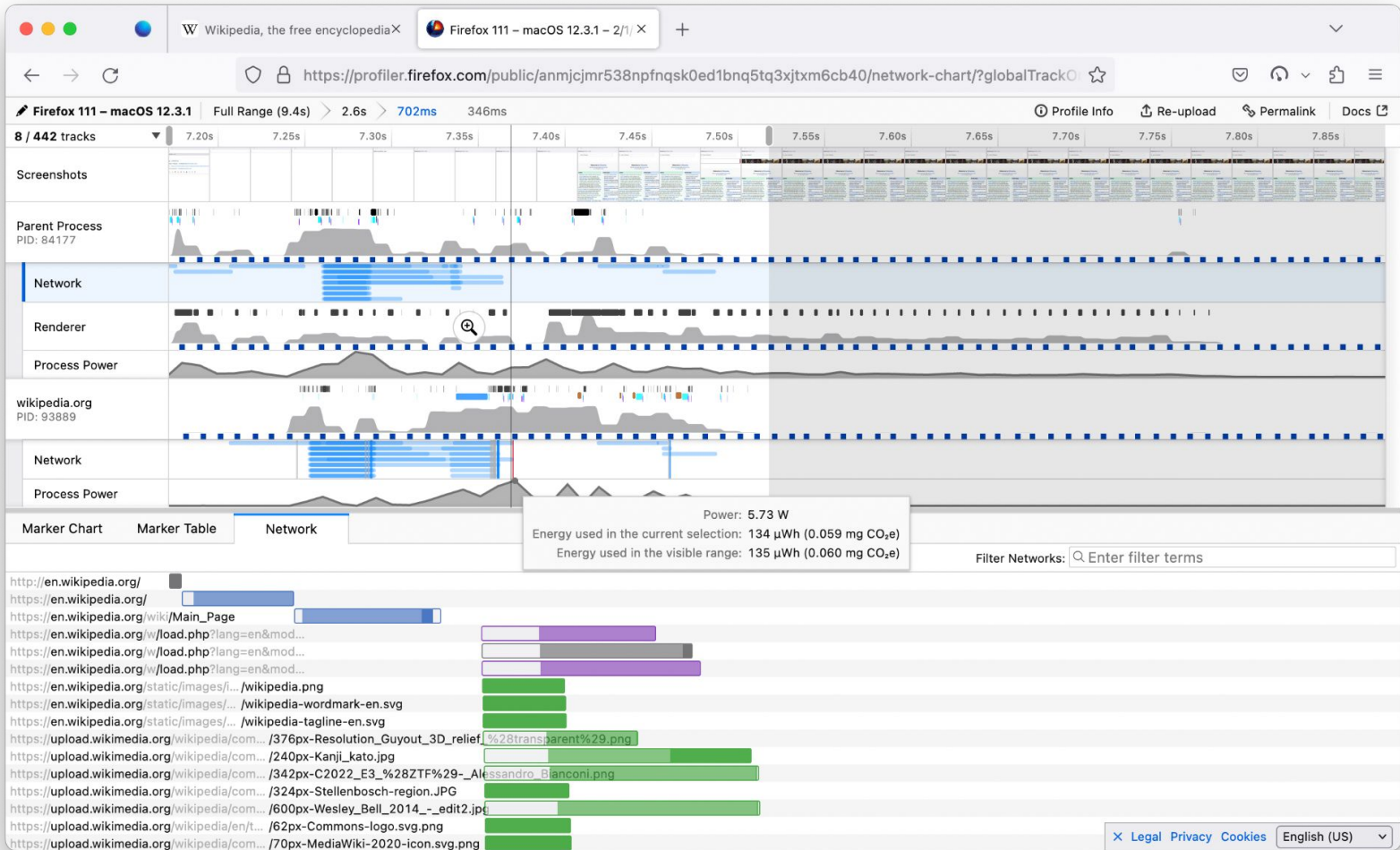
The image shows a screenshot of the Chrome DevTools Performance tab, specifically the 'Profile Information' panel. The panel is titled 'Profile Information' and displays the following details:

- 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

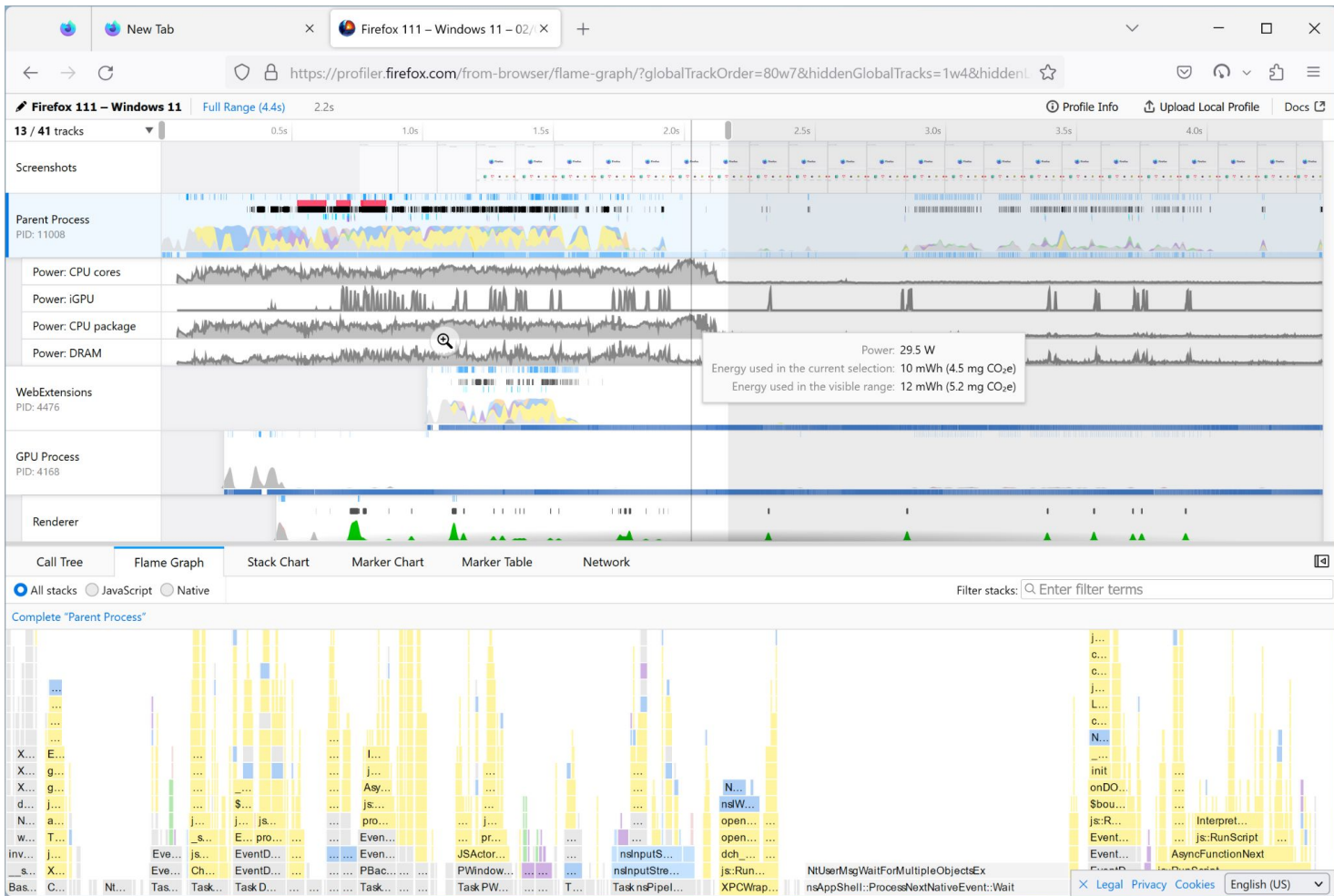




Examples



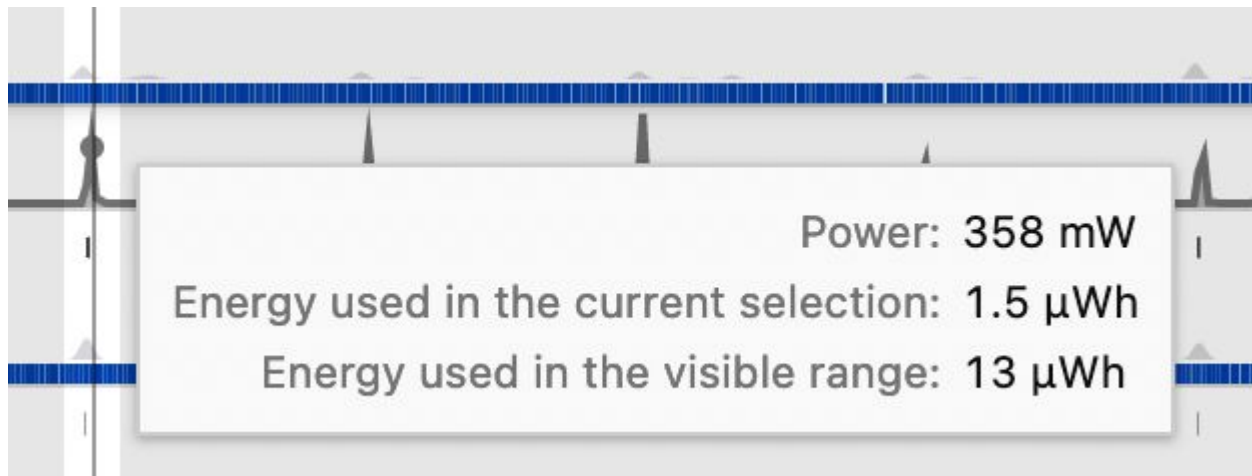
Loading Wikipedia homepage - <https://share.firefox.dev/3RqH4Ke>



Starting Firefox - <https://share.firefox.dev/3XOPHMP>

Blinking the cursor

Measure tiny things:



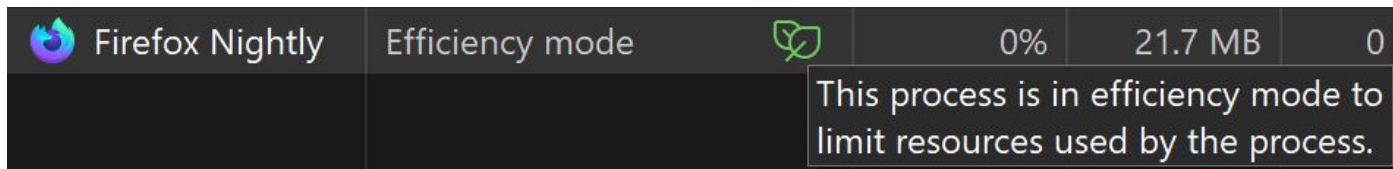
Ever wondered how much power it takes to blink the caret in the address bar?

Now you can know!

<https://share.firefox.dev/3U8hLgp>

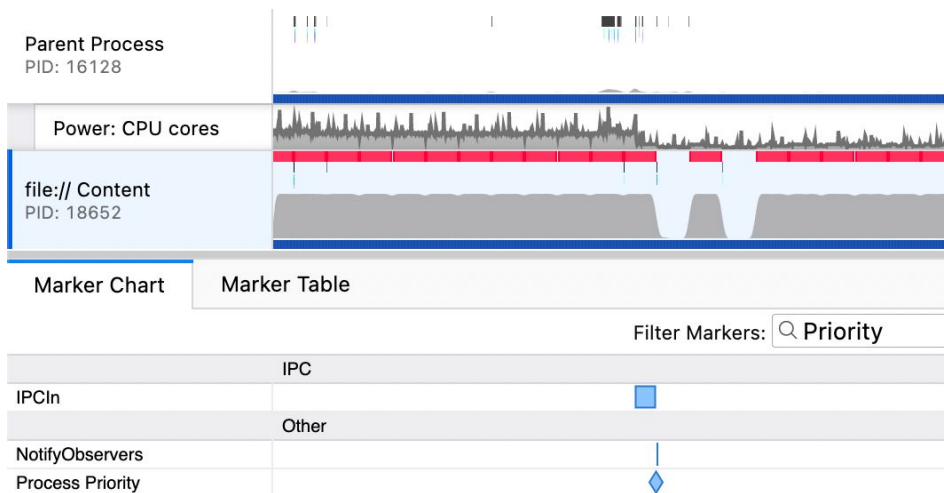


Using efficiency mode on Windows 11



Firefox Nightly Efficiency mode 0% 21.7 MB 0

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

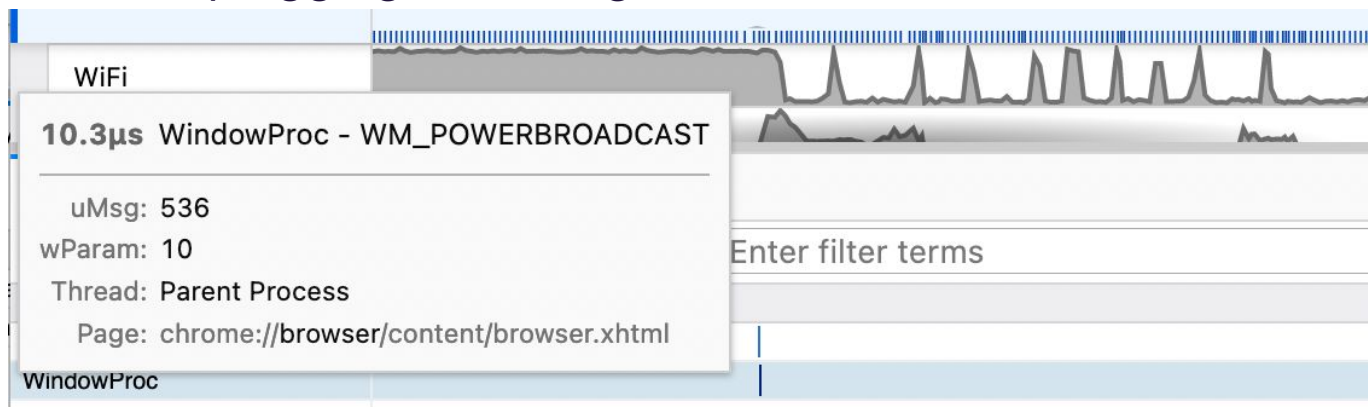


The power used by using 100% of a core drops from 10W to 2W.



Wifi power use

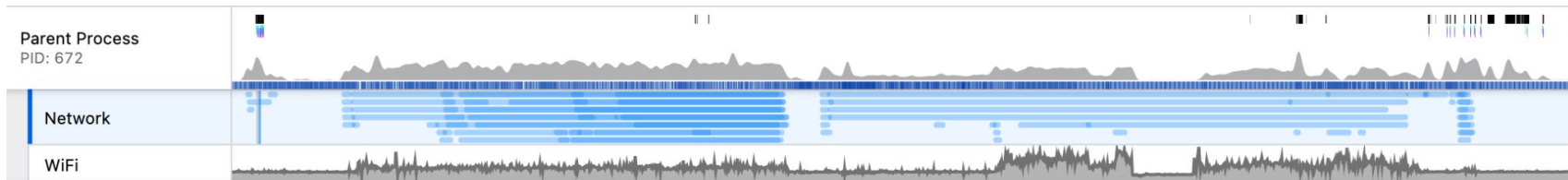
When unplugging the charger...



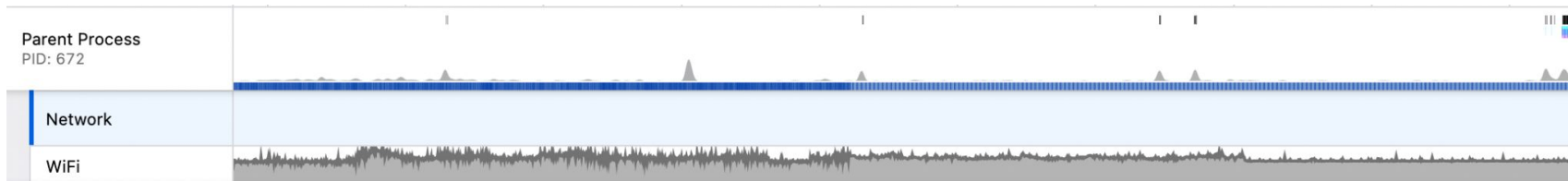
Wifi power use

Running speedtest.net

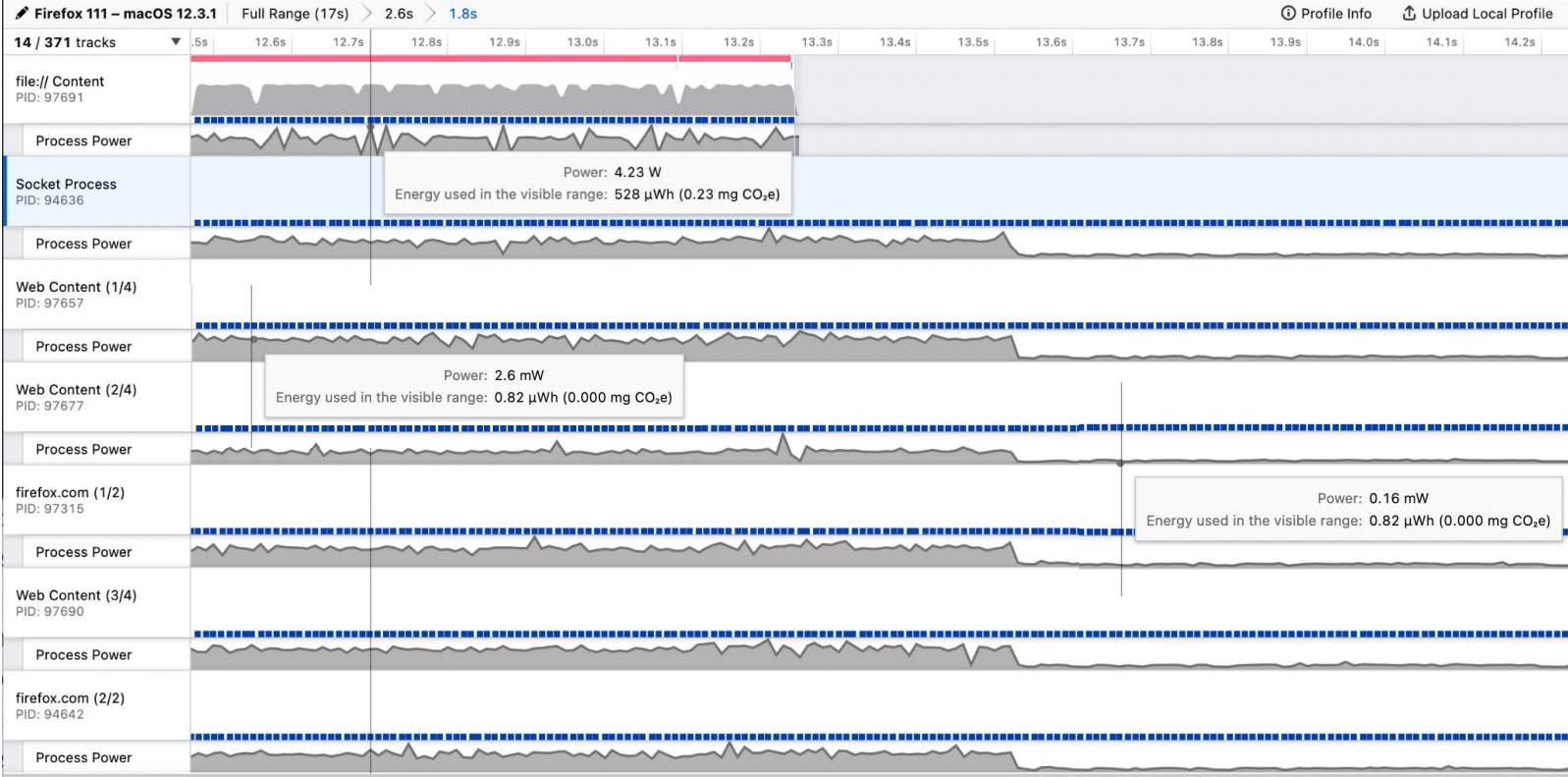
- On this machine:



- And on another machine:



Profiler overhead and CPU load



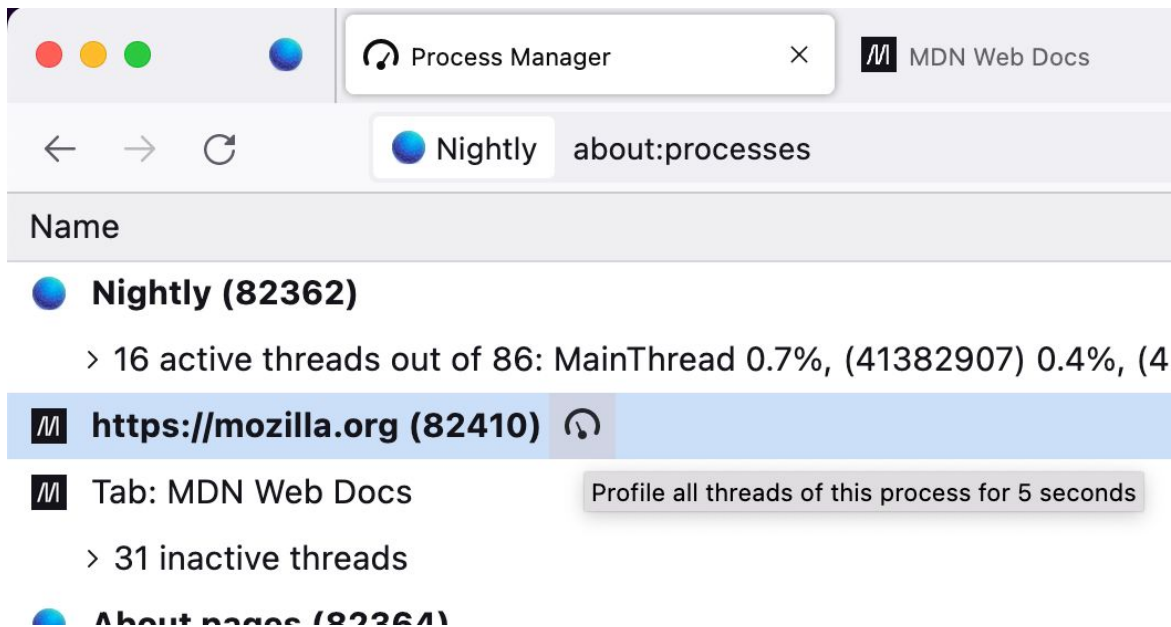


Bonus

**A few more
things**

Firefox task manager

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



The screenshot shows the Firefox Task Manager interface. At the top, there are window control buttons and a title bar with 'Process Manager' and 'MDN Web Docs'. Below that is a navigation bar with back, forward, and refresh icons, and a search bar containing 'Nightly' and 'about:processes'. The main area displays a list of processes:

- Nightly (82362)**
 - > 16 active threads out of 86: MainThread 0.7%, (41382907) 0.4%, (4
- https://mozilla.org (82410)** (highlighted in blue) with a power icon (⚡) and a tooltip that says "Profile all threads of this process for 5 seconds".
- Tab: MDN Web Docs**
 - > 31 inactive threads
- About pages (82364)**



CO₂ equivalent

Power: 2.68 W

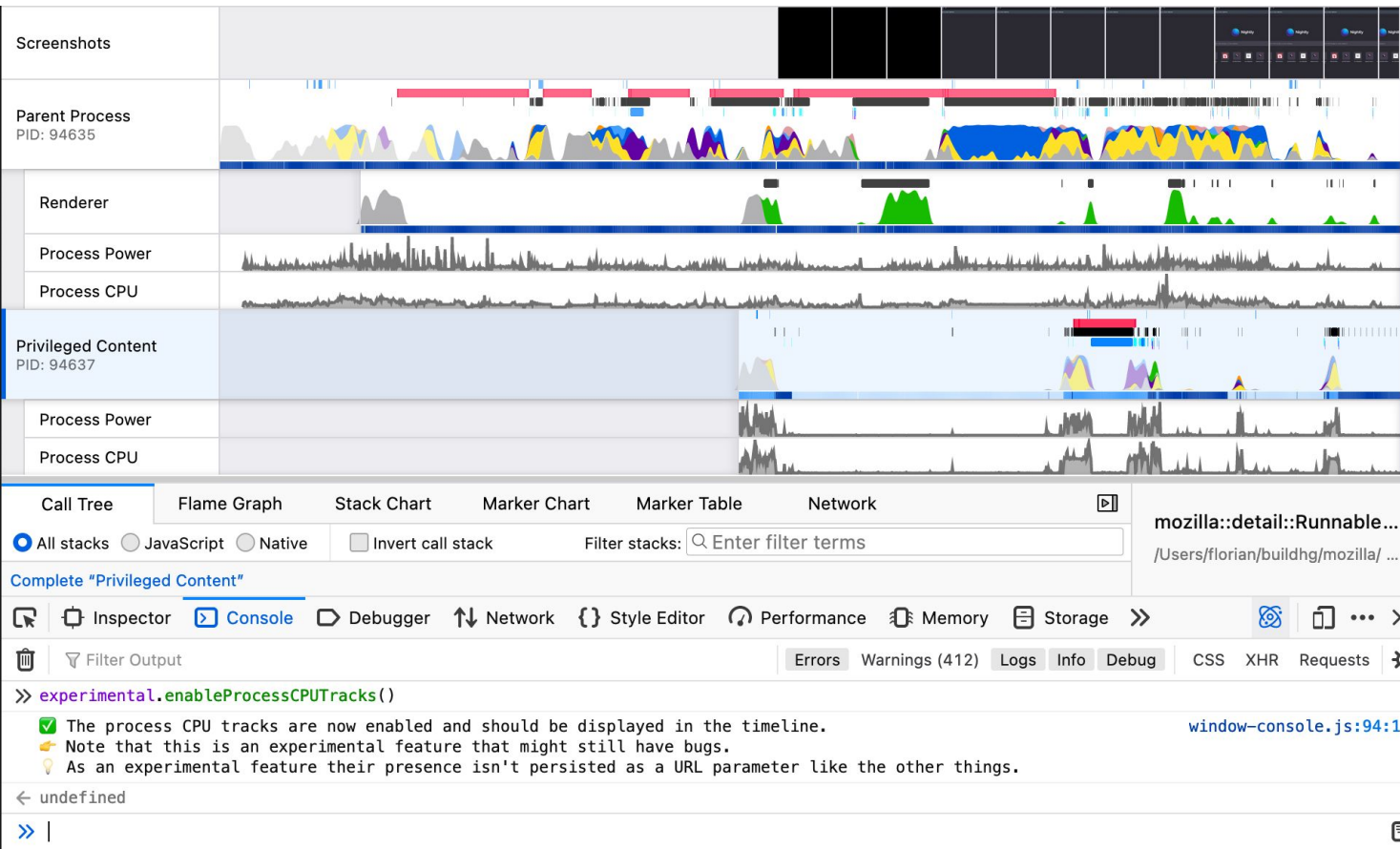
Energy used in the current selection: 63 μ Wh (0.028 mg CO₂e)

Energy used in the visible range: 168 μ Wh (0.074 mg CO₂e)

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



Process CPU



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

Thanks! Questions?

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

