Using the Firefox Profiler for performance analysis

profiler.firefox.com

Julien Wajsberg
with a lot of content coming from Nazım Can Altınova
Slides: https://share.firefox.dev/profiler-intro-fosdem-2023
FOSDEM 2023 - February 2023
Key takeaways from this talk:

- What is a profiler?
- How do I capture and share a profile?
- How do I analyze a profile?
profiler.firefox.com

github.com/firefox-devtools/profiler/
What is a profiler?
Where can we measure performance?

- Locally, on the developer’s computer
- In CI, with automated tests
- Real User Monitor (measure on your user’s computer)
Where is performance measured?

- Locally on the developer’s computer
- In CI with automated tests
- Real User Monitor (aka measure on your user’s computer)
Helps developers analyze performance issues
Gives insight and clues

This is detective work
🤔 Do not guess
⏰ Measure
How to capture a performance profile?
(live demo, skip 7 slides forward)
1. Go to profiler.firefox.com
1. Go to [profiler.firefox.com](https://profiler.firefox.com)
2. Follow the instructions
Recording UI

Record with: Ctrl + Shift + 1
Capture with: Ctrl + Shift + 2

Preset

More settings
https://share.firefox.dev/3HSmyyU
Share your captured profile

Share Performance Profile

Upload your profile and make it accessible to anyone with the link. This profile is from Firefox Nightly, so by default all information is included.

Include additional data that may be identifiable

- Include hidden threads
- Include hidden time range
- **Include screenshots**
- Include resource URLs and paths
- Include extension information

Download (3.79 MB)  Upload
A few rules to capture good data

- Isolate the problem as much as possible.
- Ensure you reproduced the results you want.
- Keep recording until you are happy with the captured data.
Different types of captured data
What is sampling?

Stops the execution of the program at a fixed rate and records relevant information like the current stack.
<table>
<thead>
<tr>
<th>Total (samples)</th>
<th>Self</th>
</tr>
</thead>
<tbody>
<tr>
<td>98%</td>
<td>1,780</td>
</tr>
<tr>
<td>90%</td>
<td>1,625</td>
</tr>
<tr>
<td>45%</td>
<td>807</td>
</tr>
<tr>
<td>45%</td>
<td>806</td>
</tr>
<tr>
<td>45%</td>
<td>806</td>
</tr>
<tr>
<td>44%</td>
<td>805</td>
</tr>
<tr>
<td>44%</td>
<td>797</td>
</tr>
<tr>
<td>44%</td>
<td>796</td>
</tr>
<tr>
<td>44%</td>
<td>795</td>
</tr>
<tr>
<td>42%</td>
<td>751</td>
</tr>
<tr>
<td>42%</td>
<td>751</td>
</tr>
<tr>
<td>42%</td>
<td>751</td>
</tr>
<tr>
<td>41%</td>
<td>750</td>
</tr>
<tr>
<td>35%</td>
<td>642</td>
</tr>
<tr>
<td>35%</td>
<td>642</td>
</tr>
<tr>
<td>34%</td>
<td>615</td>
</tr>
<tr>
<td>34%</td>
<td>615</td>
</tr>
<tr>
<td>34%</td>
<td>614</td>
</tr>
<tr>
<td>34%</td>
<td>612</td>
</tr>
<tr>
<td>33%</td>
<td>591</td>
</tr>
<tr>
<td>33%</td>
<td>591</td>
</tr>
<tr>
<td>33%</td>
<td>591</td>
</tr>
<tr>
<td>32%</td>
<td>581</td>
</tr>
<tr>
<td>0,3%</td>
<td>5</td>
</tr>
<tr>
<td>0,1%</td>
<td>2</td>
</tr>
<tr>
<td>0,1%</td>
<td>1</td>
</tr>
<tr>
<td>0,1%</td>
<td>1</td>
</tr>
<tr>
<td>0,7%</td>
<td>13</td>
</tr>
<tr>
<td>0,3%</td>
<td>5</td>
</tr>
<tr>
<td>0,2%</td>
<td>3</td>
</tr>
</tbody>
</table>

- **EventListener.handleEvent**
- **EventDispatcher#handleEvent**
- **EventDispatcher#dispatchEvent**
- **EventDispatcher#dispatchEventWithCapture**
- **EventDispatcher#dispatchEventForPluginEvent**
- **EventDispatcher#batchedUpdates**
- **EventDispatcher#batchedUpdates$1**
- **flushSyncCallbacksOnlyInLegacyMode**
- **flushSyncCallbacks**
- **self-hosted:1406.24**
- **performSyncWorkOnRoot**
- **commitRoot**
- **commitRootImpl**
- **commitLayoutEffects**
- **commitLayoutEffects._begin**
- **commitLayoutEffects._complete**
- **commitLayoutEffectOn Fiber**
- **componentDidUpdate**
- **_scheduleDraw**
- **_doDrawCanvas**
- **timeCode**
- **console.log**
- **Performance.measure**
- **Performance.now**
- **Performance.mark**
- **componentDidUpdate**
- **commitHookEffectListMount**
- **componentDidMount**
Sampling provide a view into the profiled program
Short but important events can be missed
Markers to the rescue!
What are markers?

Markers provide a view into the executing code that does not miss anything.
Limits

● Measuring on your computer doesn’t mean it’s always the same on every computer.
● The profiling itself can skew the results.
Firefox Profiler analysis UI
(live demo, skip to “advanced topics”)

27
Range selection
Screenshots in the timeline

4 / 16 tracks

Screenshots

Parent Process
PID: 36282

Renderer

localhost
PID: 43725
Category graph in the timeline

<table>
<thead>
<tr>
<th>4 / 16 tracks</th>
<th>0.2s</th>
<th>0.4s</th>
<th>0.6s</th>
<th>0.8s</th>
<th>1.0s</th>
<th>1.2s</th>
<th>1.4s</th>
<th>1.6s</th>
<th>1.8s</th>
</tr>
</thead>
<tbody>
<tr>
<td>Screenshots</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parent Process</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
PID: 38282    |      |      |      |      |      |      |      |      |      |
| Renderer      |      |      |      |      |      |      |      |      |      |
| localhost     |      |      |      |      |      |      |      |      |      |
PID: 43725    |      |      |      |      |      |      |      |      |      |
Markers in the timeline

<table>
<thead>
<tr>
<th>4 / 16 tracks</th>
<th>0.2s</th>
<th>0.4s</th>
<th>0.6s</th>
<th>0.8s</th>
<th>1.0s</th>
<th>1.2s</th>
<th>1.4s</th>
<th>1.6s</th>
<th>1.8s</th>
</tr>
</thead>
<tbody>
<tr>
<td>Screenshots</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parent Process</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PID: 36282</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Renderer</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>localhost</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PID: 43725</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Marker tooltips

14.1ms mousemove - DOMEvent
Event Target: `canvas` at `5d6570` class="chartCanvas flameGraphCanvas hover"
Latency: 3.921ms
Event Type: mousemove
Thread: http://localhost
Page: http://localhost:4242/public/48hra4hvn9n9g2br1mn3xsmxcaetb2kctvja0fimage-graph?globalTrackOrder=0
wjd&hiddenGlobalTracks=0w26&hiddenLocalTracksByPid=11281-0w9&range=9564m3470&thread=9&transforms=df-20&v=8

276μs Reflow
Type: Paint
Thread: http://localhost
Page: http://localhost:4242/public/48hra4hvn9n9g2br1mn3xsmxcaetb2kctvja0fimage-graph?globalTrackOrder=0
wjd&hiddenGlobalTracks=0w26&hiddenLocalTracksByPid=11281-0w9&range=9564m3470&thread=9&transforms=df-20&v=8
Stack: First invalidated 14.670ms before the flush, at:
- Reflow http://localhost:4242/public/48hra4hvn9n9g2br1mn3xsmxcaetb2kctvja0fimage-graph?globalTrackOrder=0
wjd&hiddenGlobalTracks=0w26&hiddenLocalTracksByPid=11281-0w9&range=9564m3470&thread=9&transforms=df-20&v=8

894ms CSS animation
Name: `tab-throbber-animation`
Animated Properties: transform
Can Run on Compositor: true
Target: `hbox@7f679653a80` class="tab-throbber"
Thread: Parent Process
Page: chrome://browser/content/browser.xhtml
Debugging with the profiler

- The profiler isn’t only for profiling!
- Get a quick overview of the program flow for some code you don’t know
  - Codepaths, file accesses, network requests, various events
- Instrument your JS code with the Performance Timing API to add more data that will show up in profilers
Advanced topics
Memory allocation instrumentation

https://profiler.firefox.com/docs/#/./memory-allocations
Chrome (and node) profile importer
Compare profiles
Conclusion
Measure, don’t guess.
This is a tool for your toolbox.
Use the profiler to debug.
Share profiles with your team.
Thank you

Firefox Profiler
https://profiler.firefox.com/

Firefox Profiler Documentation:
https://profiler.firefox.com/docs/

Matrix channel:
https://chat.mozilla.org/#/room/#profiler:mozilla.org

Slides:
https://share.firefox.dev/profiler-intro-fosdem-2023