EN
English

Universidad de Oviedo

School of Computer Science

SOFTWARE ARCHITECTURE

Software Architecture

Lab. 12

Monitoring & profiling

How-to do a presentation

Jose Emilio Labra Gayo
Pablo González
Irene Cid
Hugo Lebredo

2019-20

Monitoring and profiling

Monitoring: Observe the behaviour at runtime while software is running

- Dashboards
- Usually, after deployment

Profiling: Measure performance of a software while it is running

- Identify parts of a system that contribute to a performance problem
- Show where to concentrate the efforts
- Usually before deployment

Monitoring & profiling

Monitors an application while it is running

Records performance (CPU & memory usage)

JavaScript:

- Chrome (Timeline), Firefox Developer Edition (Performance tool), Microsoft(Ajax View)

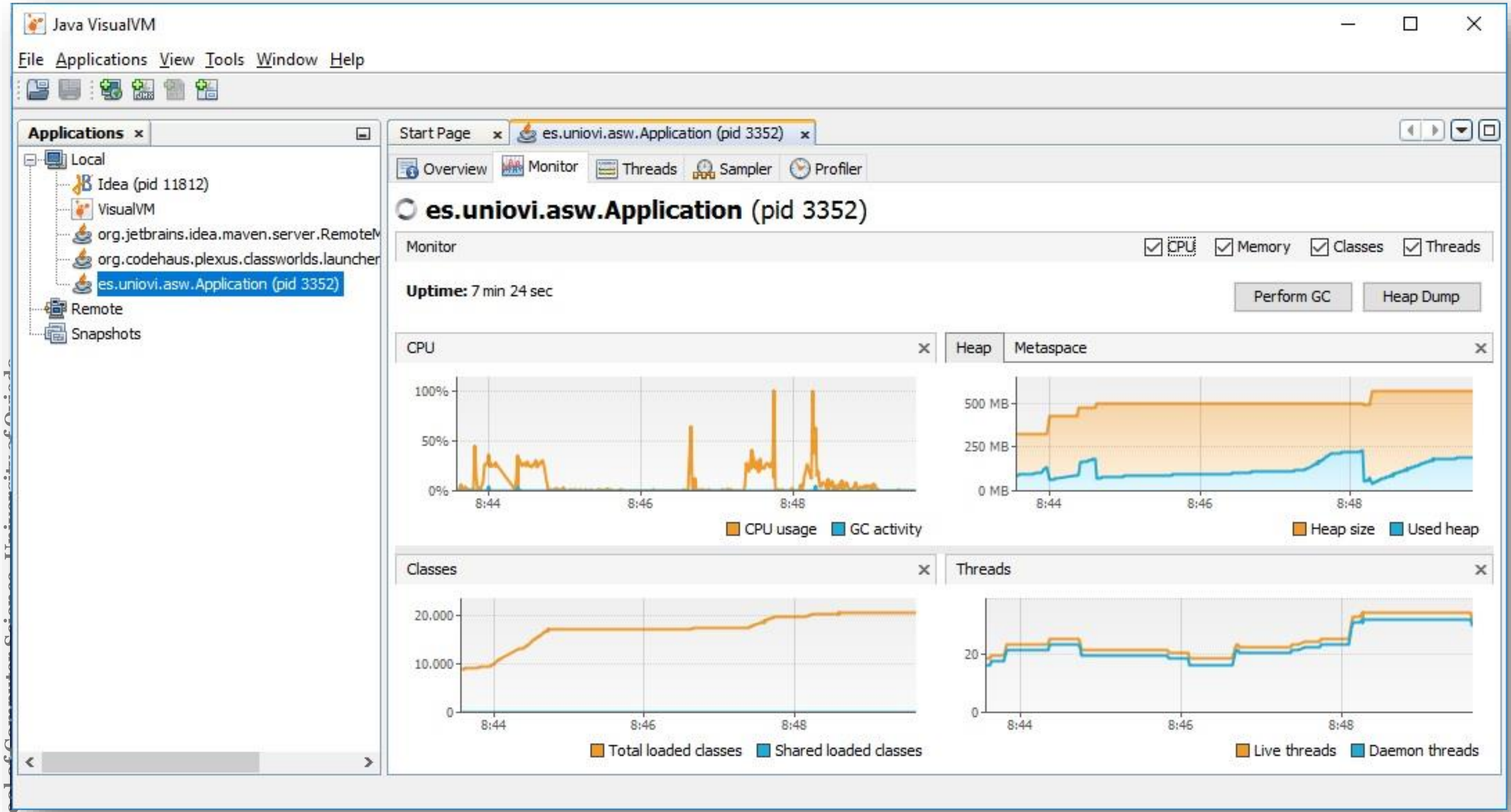
Server-side:

- JVisualVM, JProfiler, YourKit, JConsole
- Monitoring: Graphite, Datadog

VisualVM

- https://visualvm.github.io/
- `jvisualvm`

Java/server JVisualVM



Local
- Idea (pid 11812)
- VisualVM
- org.jetbrains.idea.maven.server.RemoteM
- org.codehaus.plexus.classworlds.launcher
- es.uniovi.asw.Application (pid 3352)

Remote

Snapshots

es.uniovi.asw.Application (pid 3352)

Monitor ☑ CPU ☑ Memory ☑ Classes ☑ Threads

Uptime: 7 min 24 sec

🟧 CPU usage 🟦 GC activity

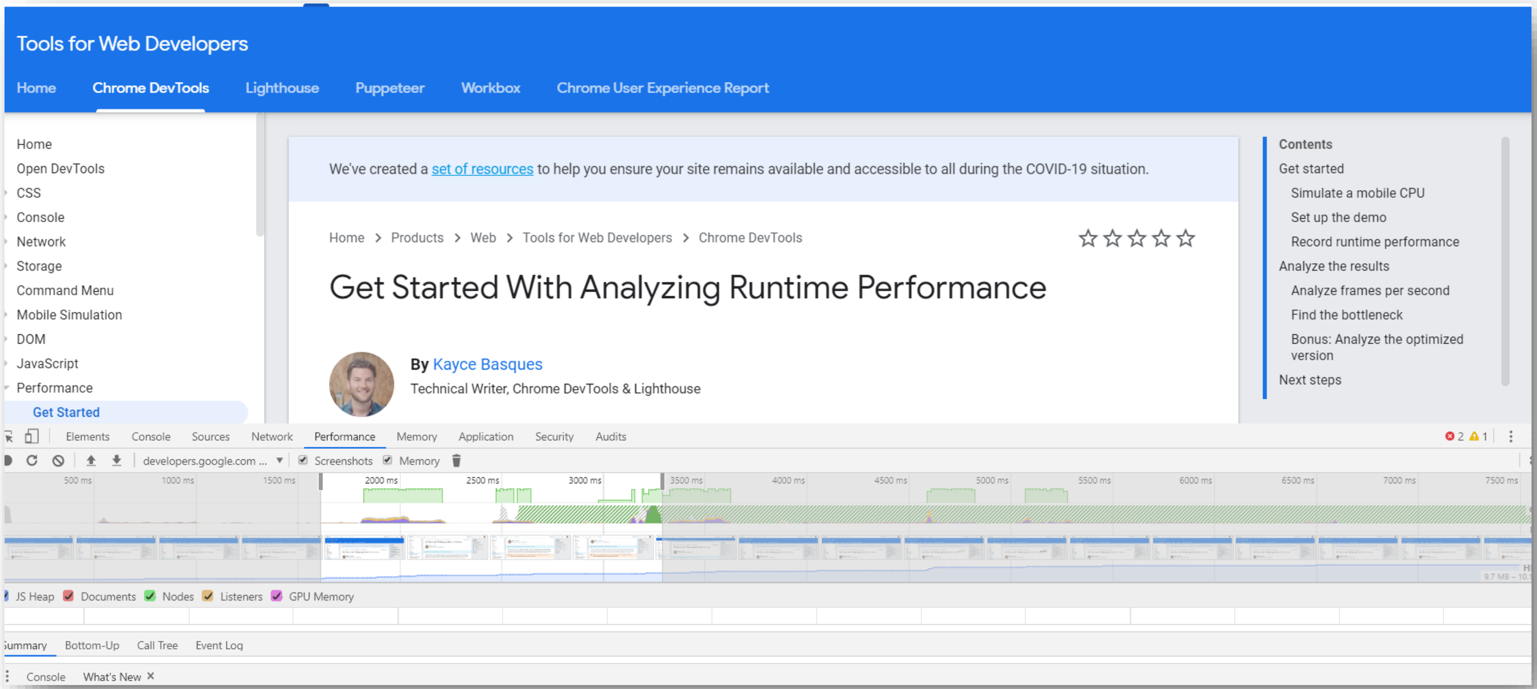
🟧 Heap size 🟦 Used heap

🟧 Total loaded classes 🟦 Shared loaded classes

🟧 Live threads 🟦 Daemon threads

Browser: developer tools

Monitor/check performance



We've created a set of resources to help you ensure your site remains available and accessible to all during the COVID-19 situation.

Home > Products > Web > Tools for Web Developers > Chrome DevTools

Get Started With Analyzing Runtime Performance

By Kayce Basques
Technical Writer, Chrome DevTools & Lighthouse

Contents
- Get started
  - Simulate a mobile CPU
  - Set up the demo
  - Record runtime performance
- Analyze the results
  - Analyze frames per second
  - Find the bottleneck
  - Bonus: Analyze the optimized version
- Next steps

https://developers.google.com/web/tools/chrome-devtools/evaluate-performance

Example with Google Chrome

Incognito mode

At the top right, click the three dots and then New Incognito Window.

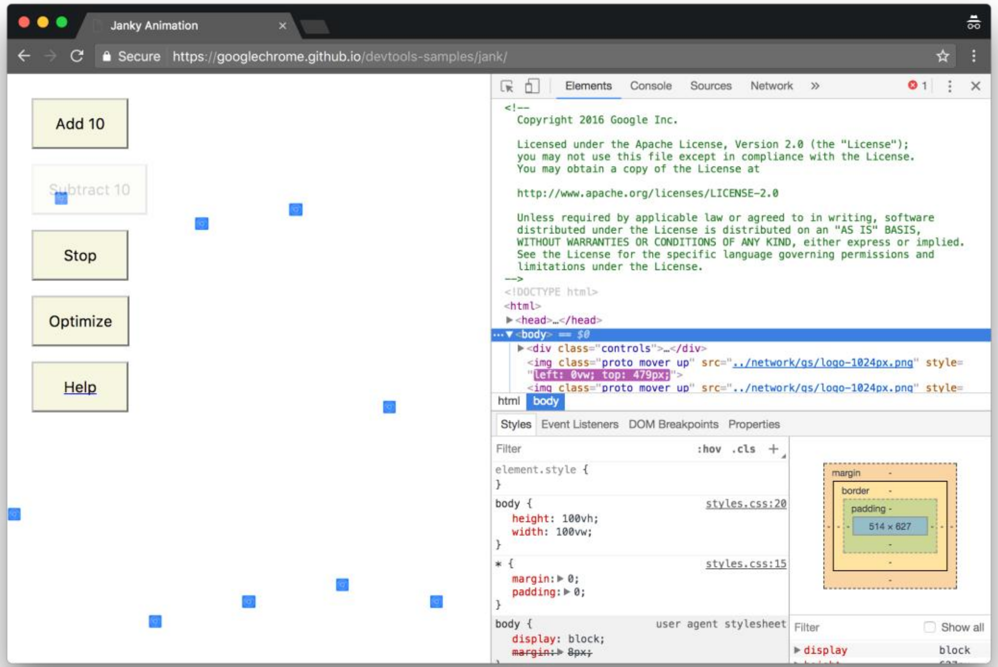
Windows, Linux, or Chrome OS: Press Ctrl + Shift + n.
Mac: Press ⌘ + Shift + n.

DevTools

Windows, Linux: Control+Shift+I
Mac: Command+Option+I

Example with Google Chrome

https://googlechrome.github.io/devtools-samples/jank/

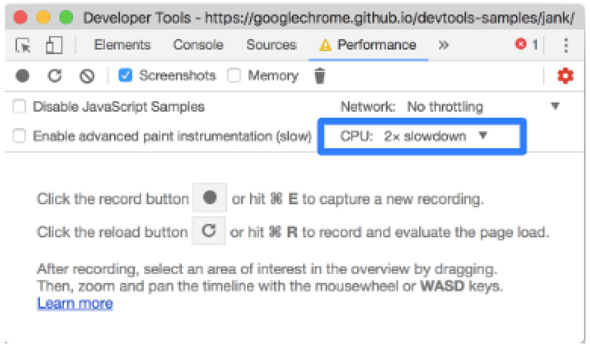


```
<!--
  Copyright 2016 Google Inc.

  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

  http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License.
-->
<!DOCTYPE html>
<html>
▶<head>…</head>
…▼<body> == $0
  ▶<div class="controls">…</div>
   <img class="proto mover up" src="../network/gs/logo-1024px.png" style=
   "left: 0vw; top: 479px;">
   <img class="proto mover up" src="../network/gs/logo-1024px.png" style=
```

```
element.style {
}
body {                                   styles.css:20
  height: 100vh;
  width: 100vw;
}
* {                                      styles.css:15
  margin:▶ 0;
  padding:▶ 0;
}
body {                        user agent stylesheet
  display: block;
  ~~margin:▶ 8px;~~
```

```
▶display                block
```

Performance>CPU>2 x Slowdown


Developer Tools - https://googlechrome.github.io/devtools-samples/jank/
Elements
Console
Sources
Performance
Screenshots
Memory

☐ Disable JavaScript Samples   Network: No throttling

☐ Enable advanced paint instrumentation (slow)   CPU: 2x slowdown

Click the record button ● or hit ⌘ E to capture a new recording.

Click the reload button C or hit ⌘ R to record and evaluate the page load.

After recording, select an area of interest in the overview by dragging.
Then, zoom and pan the timeline with the mousewheel or **WASD** keys.
Learn more

Performance>Record
    click Add 10 (20 times)
    try Optimize / Un-optimize
Stop

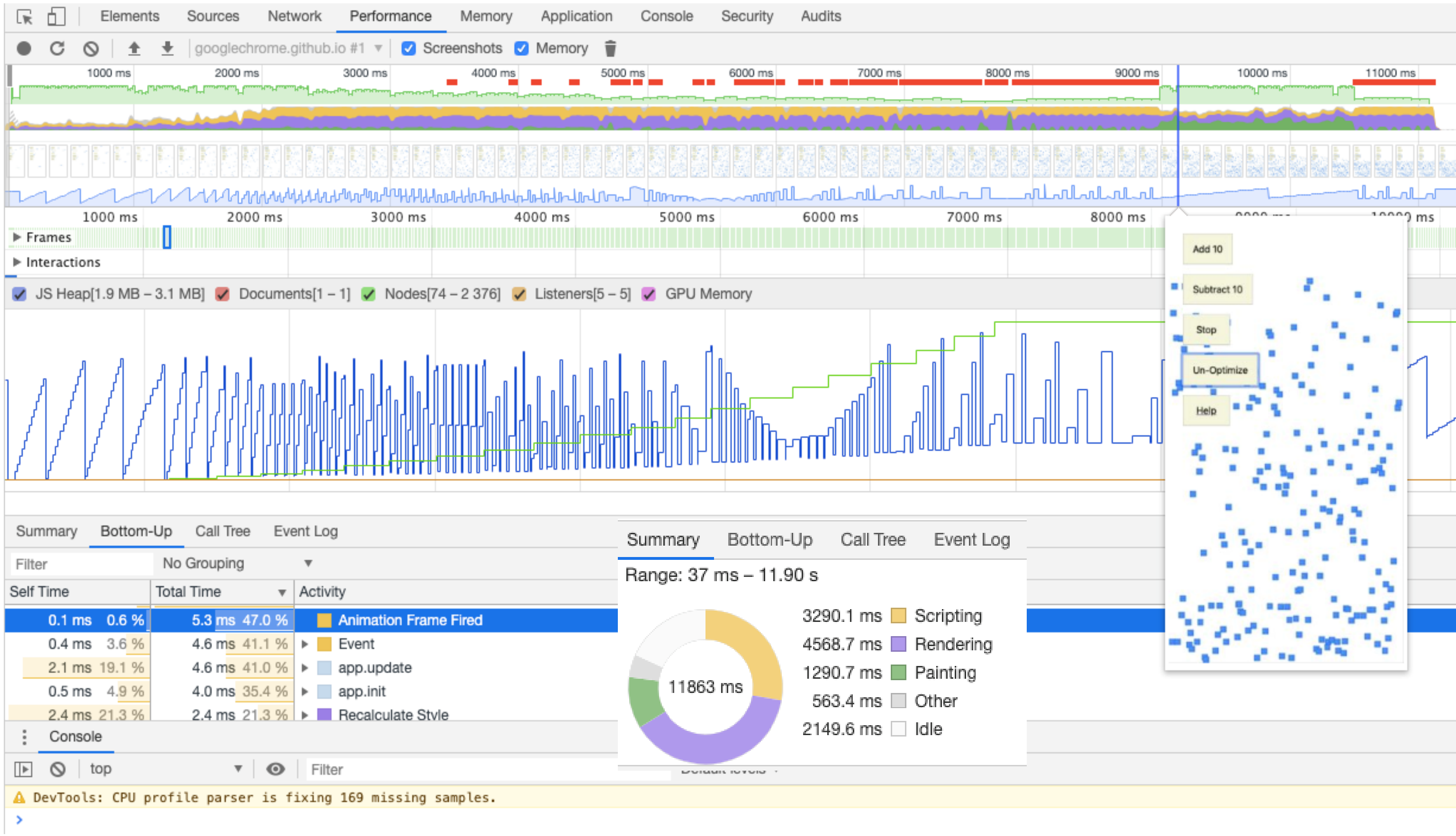
Example with Google Chrome

Profile result:

Frames per Second →

CPU →

Bottleneck→



Elements Sources Network Performance Memory Application Console Security Audits

googlechrome.github.io #1 ✓ Screenshots ✓ Memory

1000 ms 2000 ms 3000 ms 4000 ms 5000 ms 6000 ms 7000 ms 8000 ms 9000 ms 10000 ms 11000 ms

Frames

Interactions

✓ JS Heap[1.9 MB – 3.1 MB] ✓ Documents[1 – 1] ✓ Nodes[74 – 2 376] ✓ Listeners[5 – 5] ✓ GPU Memory

Summary Bottom-Up Call Tree Event Log

Filter No Grouping

Self Time		Total Time		Activity
0.1 ms	0.6 %	5.3 ms	47.0 %	Animation Frame Fired
0.4 ms	3.6 %	4.6 ms	41.1 %	Event
2.1 ms	19.1 %	4.6 ms	41.0 %	app.update
0.5 ms	4.9 %	4.0 ms	35.4 %	app.init
2.4 ms	21.3 %	2.4 ms	21.3 %	Recalculate Style

Summary Bottom-Up Call Tree Event Log

Range: 37 ms – 11.90 s

11863 ms

3290.1 ms	🟨	Scripting
4568.7 ms	🟪	Rendering
1290.7 ms	🟩	Painting
563.4 ms	⬜	Other
2149.6 ms	⬜	Idle

Add 10 Subtract 10 Stop Un-Optimize Help

Console

top Filter

```
⚠ DevTools: CPU profile parser is fixing 169 missing samples.
```

Other tools for browser

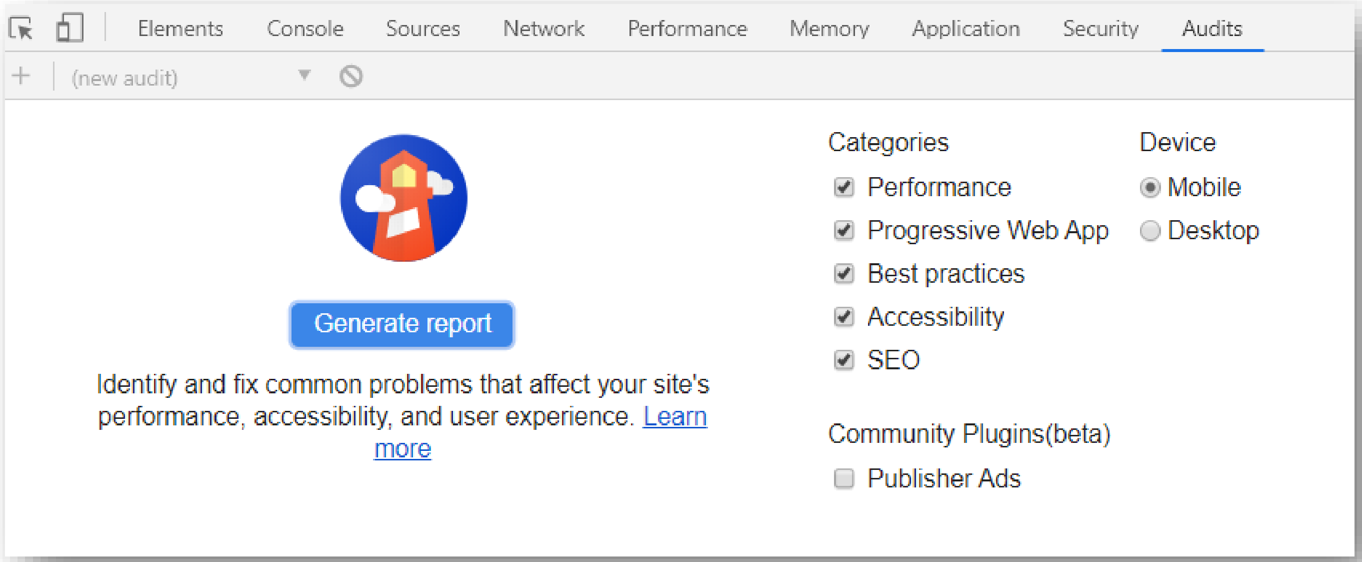
RAIL model:

Response, Animation, Idle, Load

https://developers.google.com/web/fundamentals/performance/rail

https://webpagetest.org/easy

Lighthouse (with Chrome)



Identify and fix common problems that affect your site's performance, accessibility, and user experience. Learn more

Categories
- [✓] Performance
- [✓] Progressive Web App
- [✓] Best practices
- [✓] Accessibility
- [✓] SEO

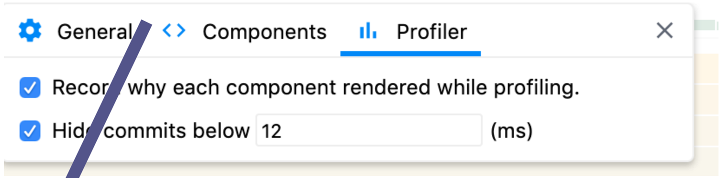
Device
- [✓ Mobile]
- Desktop

Community Plugins(beta)
- [ ] Publisher Ads

React Developer Tools

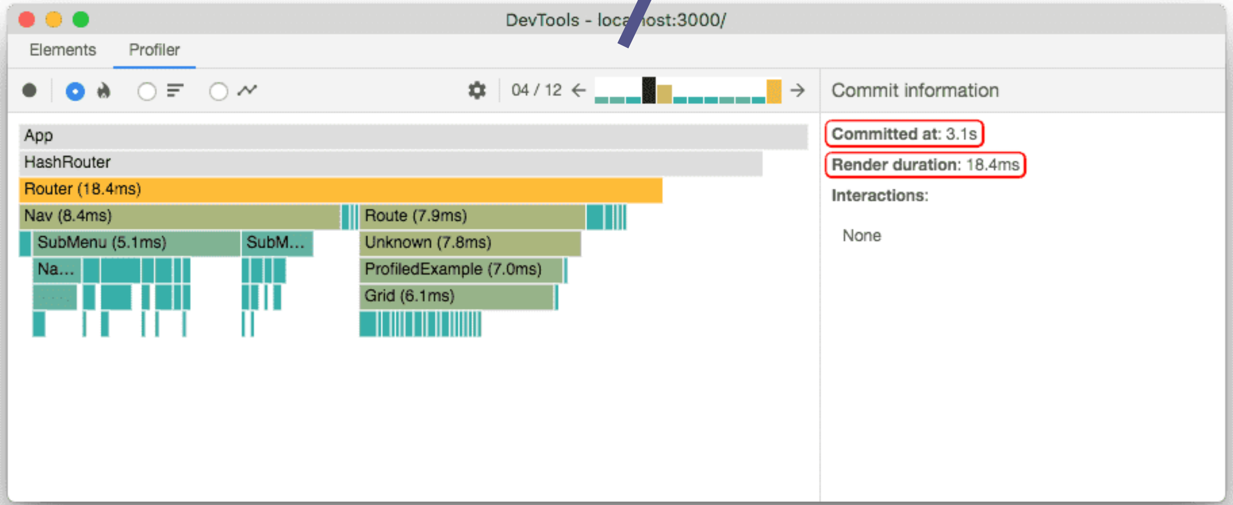
React works in two stages:

- Render
- Commit



☑ Record why each component rendered while profiling.

☑ Hide commits below 12 (ms)



Commit information

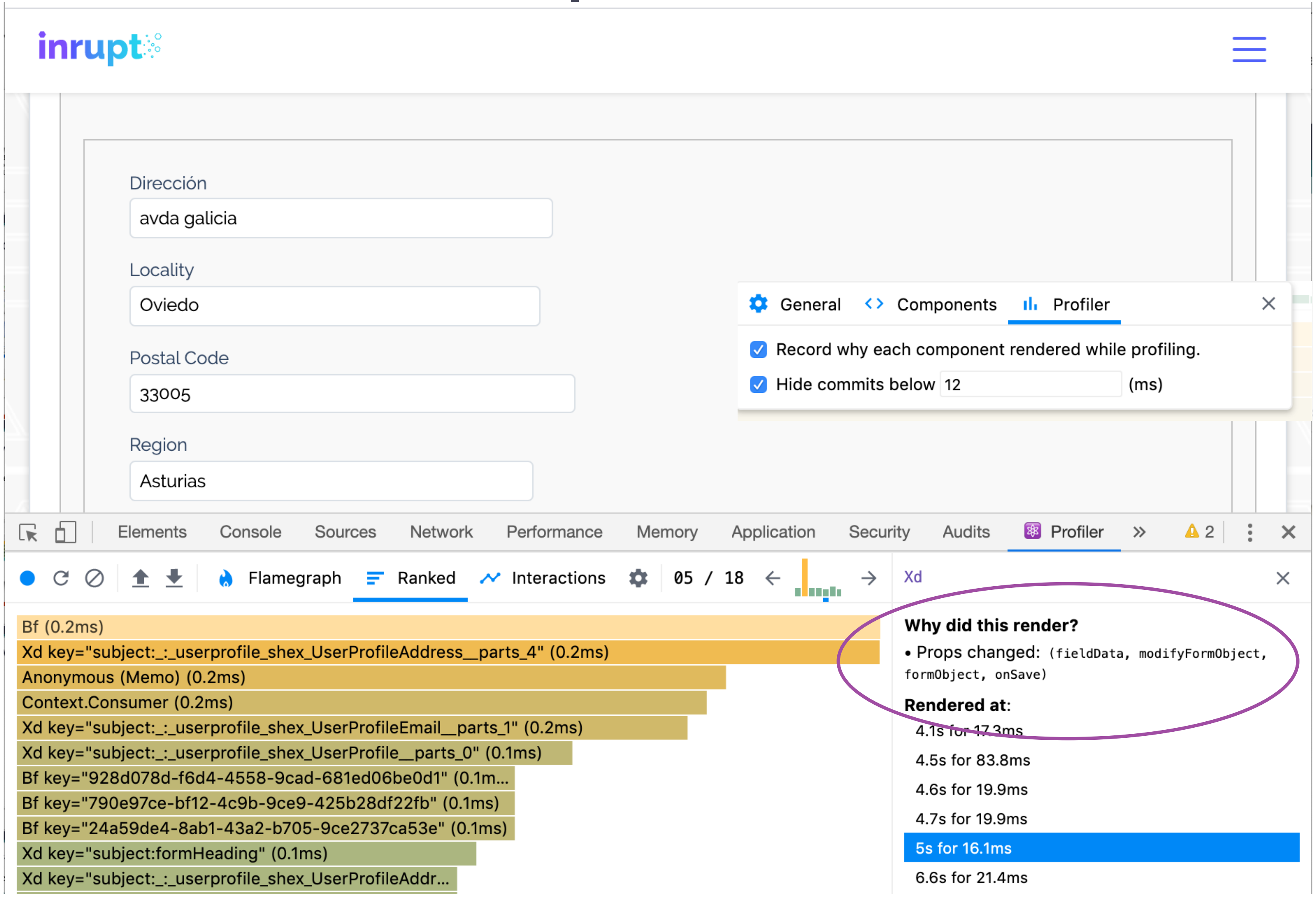
Committed at: 3.1s

Render duration: 18.4ms

Interactions:

None

React Developer Tools



Dirección: avda galicia

Locality: Oviedo

Postal Code: 33005

Region: Asturias

☑ Record why each component rendered while profiling.

☑ Hide commits below 12 (ms)

```
Bf (0.2ms)
Xd key="subject:_:_userprofile_shex_UserProfileAddress__parts_4" (0.2ms)
Anonymous (Memo) (0.2ms)
Context.Consumer (0.2ms)
Xd key="subject:_:_userprofile_shex_UserProfileEmail__parts_1" (0.2ms)
Xd key="subject:_:_userprofile_shex_UserProfile__parts_0" (0.1ms)
Bf key="928d078d-f6d4-4558-9cad-681ed06be0d1" (0.1m...
Bf key="790e97ce-bf12-4c9b-9ce9-425b28df22fb" (0.1ms)
Bf key="24a59de4-8ab1-43a2-b705-9ce2737ca53e" (0.1ms)
Xd key="subject:formHeading" (0.1ms)
Xd key="subject:_:_userprofile_shex_UserProfileAddr...
```

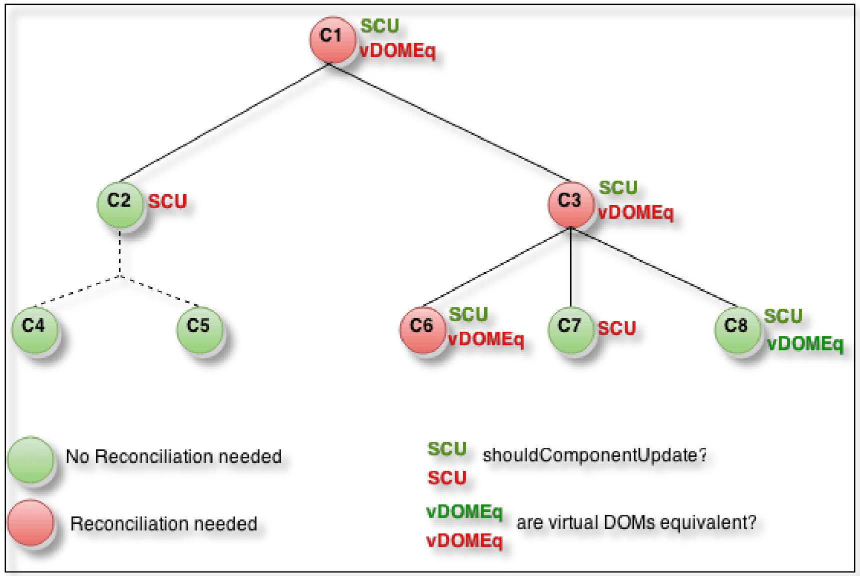
Why did this render?

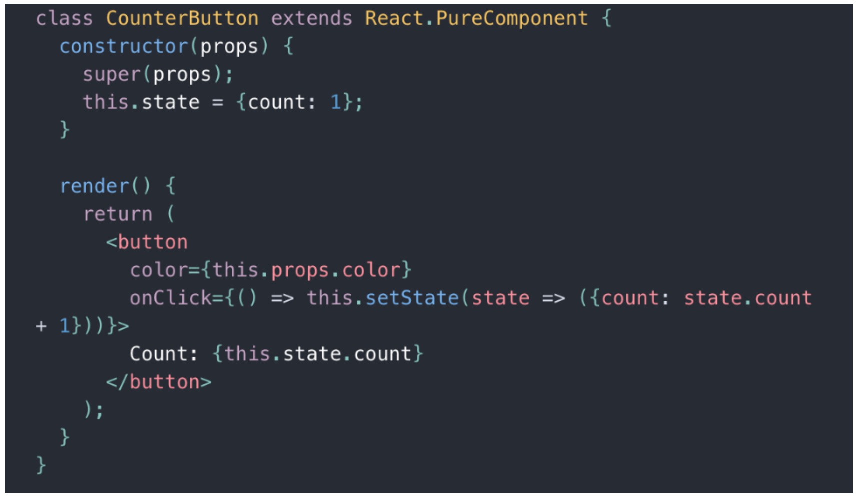
- Props changed: `(fieldData, modifyFormObject, formObject, onSave)`

Rendered at:

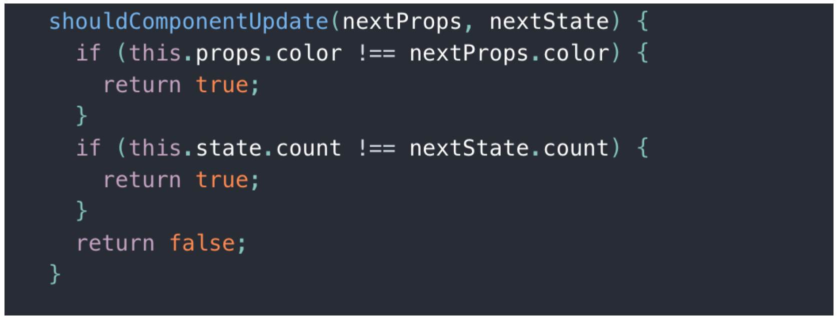
- 4.1s for 17.3ms
- 4.5s for 83.8ms
- 4.6s for 19.9ms
- 4.7s for 19.9ms
- 5s for 16.1ms
- 6.6s for 21.4ms

React DOM – Virtual DOM





```
class CounterButton extends React.PureComponent {
  constructor(props) {
    super(props);
    this.state = {count: 1};
  }

  render() {
    return (
      <button
        color={this.props.color}
        onClick={() => this.setState(state => ({count: state.count
+ 1}))}>
        Count: {this.state.count}
      </button>
    );
  }
}
```



```
shouldComponentUpdate(nextProps, nextState) {
  if (this.props.color !== nextProps.color) {
    return true;
  }
  if (this.state.count !== nextState.count) {
    return true;
  }
  return false;
}
```

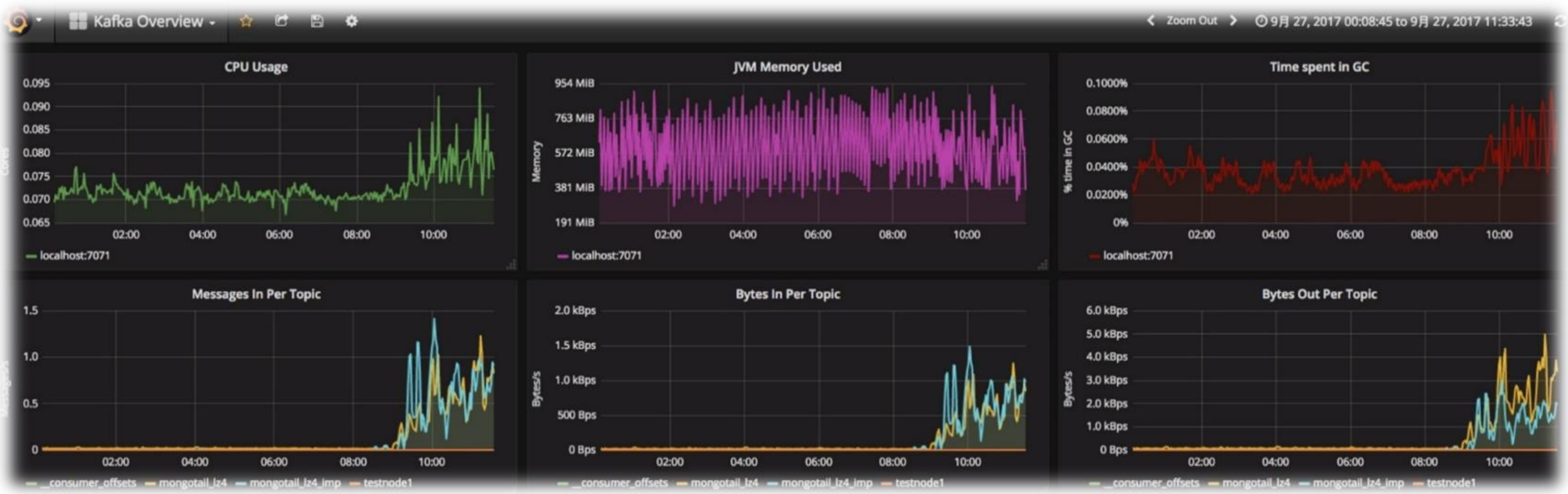
Source: https://es.reactjs.org/docs/optimizing-performance.html

Web monitoring alternatives

Spring-boot provides "Actuator" for features in production

Some systems:

- Prometheus, Graphite, Grafana, Datadog, Nagios, Sensu, …



Presenting architecture

Ideas for the presentation

Presenting architecture

For this course

- Talk about 15-20'
- Questions: 5-15'
- Teachers select the presenter

What to present?

Focus on key aspects

- Present architecture & system
- Quality attributes/strategies
- Demo
- Tests (acceptance tests, load tests,...)

Presenting architectures

DeveloperToArchitect.com

Software Architecture Monday with Mark Richards

Lesson 31 - Presenting Architecture

Mark Richards

Independent Consultant

Hands-on Software Architect / Published Author / Conference Speaker

Founder, DeveloperToArchitect.com

www.wmrichards.com

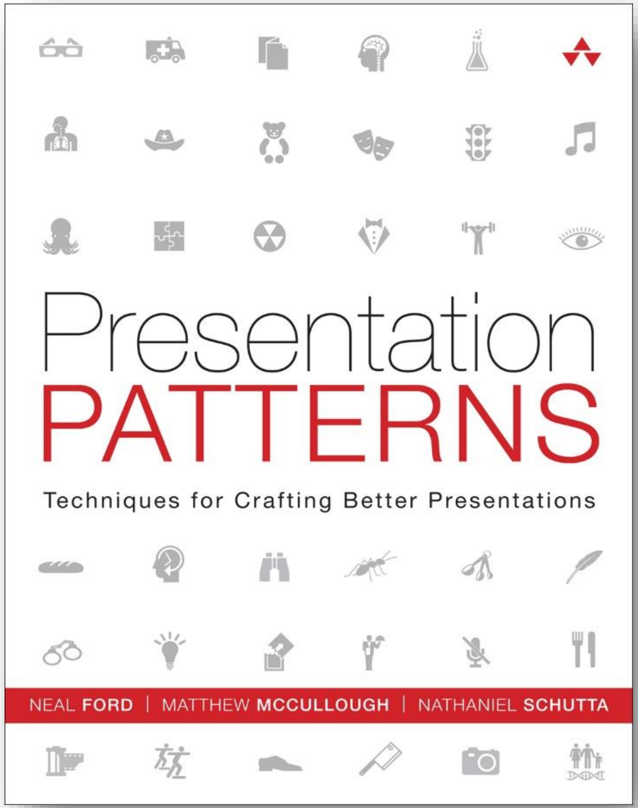
https://www.youtube.com/watch?v=pJc0l2DASpo&t=299s

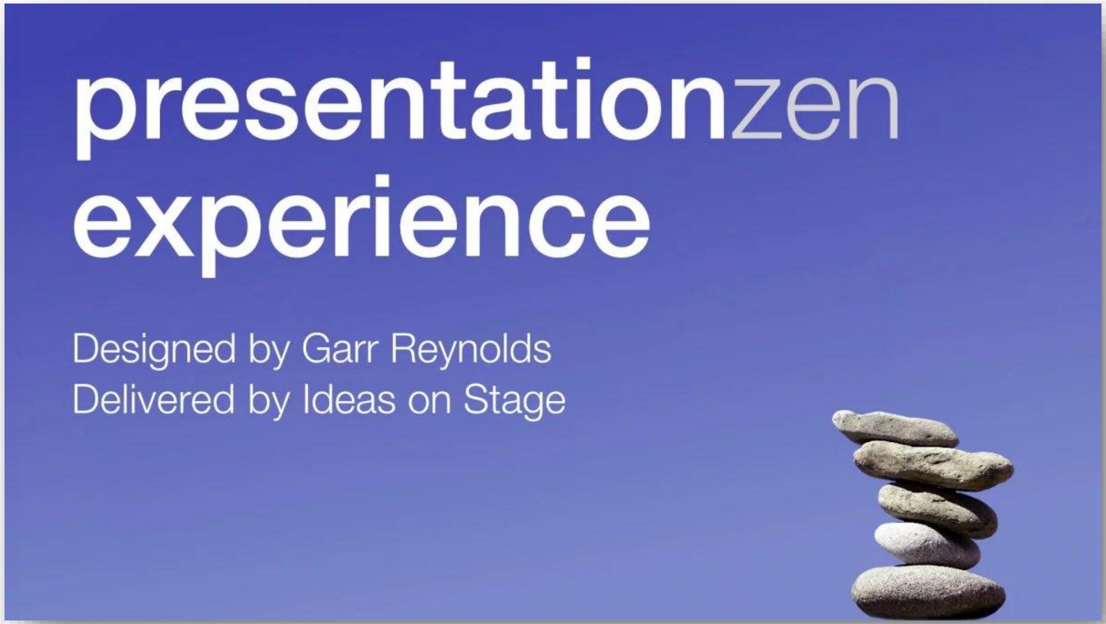
Improving presentations

https://presentationpatterns.com/resources/

https://www.oreilly.com/library/view/presentation-patterns/9781491954980/

https://www.presentationzen.com/


Presentation
PATTERNS
Techniques for Crafting Better Presentations
NEAL FORD | MATTHEW MCCULLOUGH | NATHANIEL SCHUTTA



Links

Monitoring & Profiling

Get Started With Analyzing Runtime Performance
https://developers.google.com/web/tools/chrome-devtools/evaluate-performance/

How to Use the Timeline Tool
https://developers.google.com/web/tools/chrome-devtools/evaluate-performance timeline-tool#profile-js

Presentation

Presentation Zen Garr Reynolds
https://www.presentationzen.com/
https://www.amazon.com/gp/product/0321811984