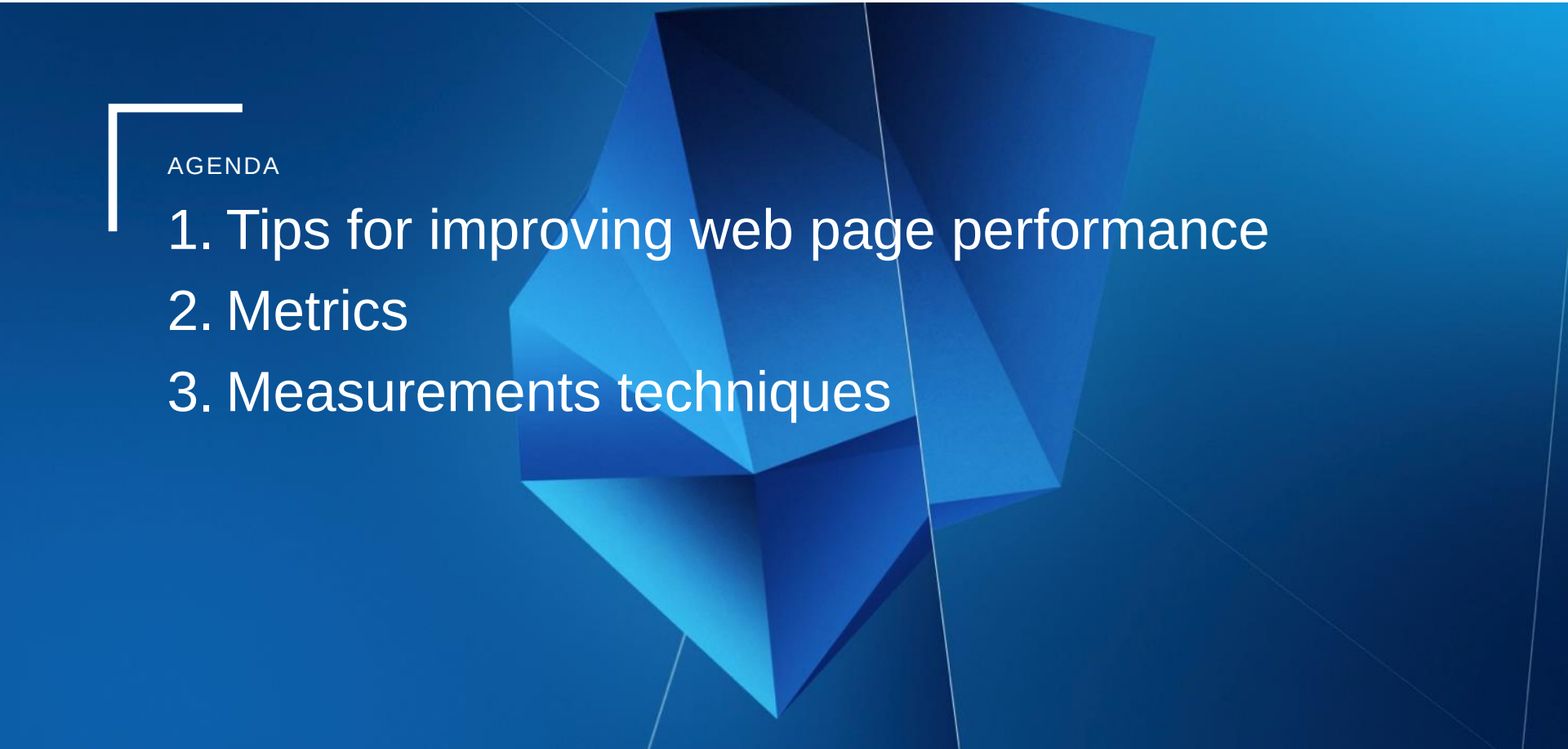
Web Application Performance From User Perspective

GFT



AGENDA

1. Tips for improving web page performance
2. Metrics
3. Measurements techniques

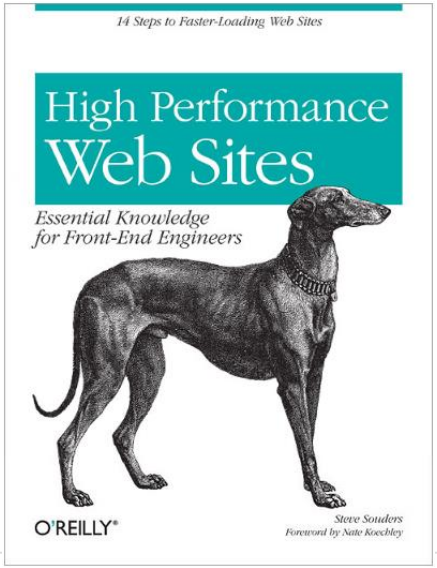
About me

GFT

- My foundation of technical expertise stems from hands on developing, testing along with team leading.
- As consultant I was a part of various test teams and was involved in testing on all software life cycle stages. I build extensive experience while testing critical functions of real time system which is GSM, financial platforms and also web and desktop applications.
- Currently I supervise testing activities in one of the project for GFT Technologies

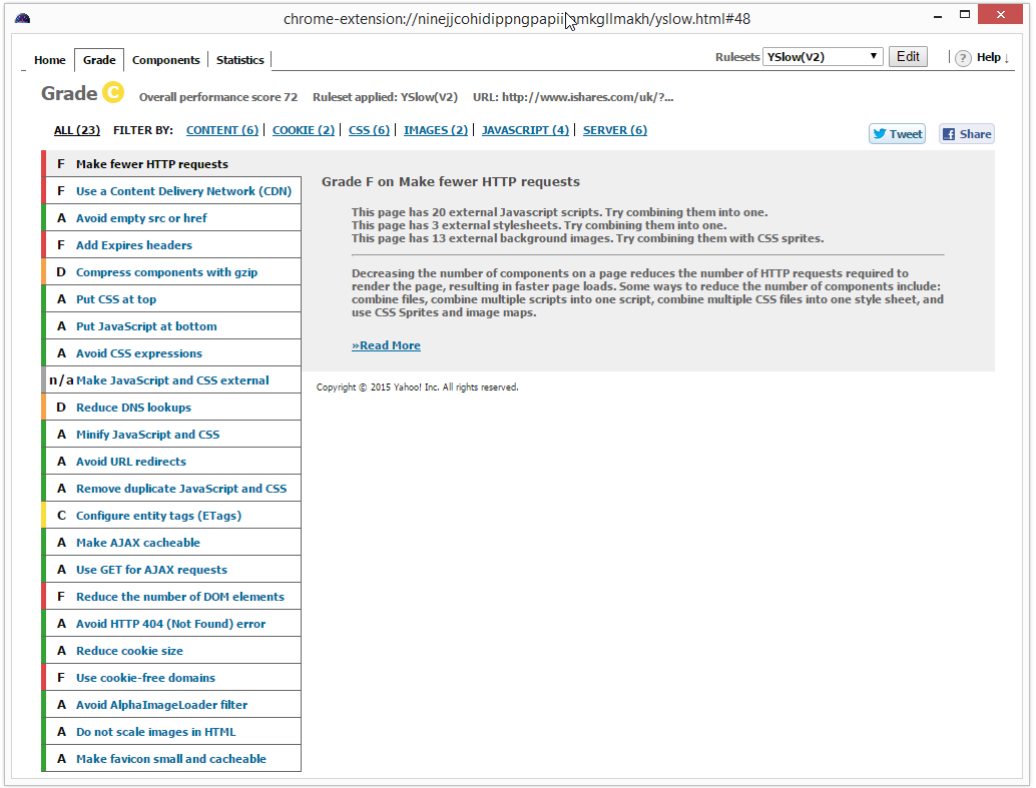
Tips For High Performance Web Sites

- Gzip Components
- Put StyleSheets at the Top
- Put Scripts at the Bottom
- Minimize HTTP Requests
- Minify JavaScript and CSS
- Make JavaScript and CSS External
- Use a Content Delivery Network
- Avoid empty src or href
- Add an Expires or a Cache-Control Header
- Avoid CSS Expressions
- Reduce DNS Lookups
- Remove Duplicate Scripts
- Make AJAX Cacheable
- Use GET for AJAX Requests
- Reduce the Number of DOM Elements
- Use Cookie-Free Domains for Components
- Do Not Scale Images in HTML
- Make favicon.ico Small and Cacheable
- Configure ETags
- Avoid Redirects
- Reduce Cookie Size
- Avoid Filters
- No 404s



Checking conformance to High Performance Web Sites Rules

- YSlow
- Google PageSpeed



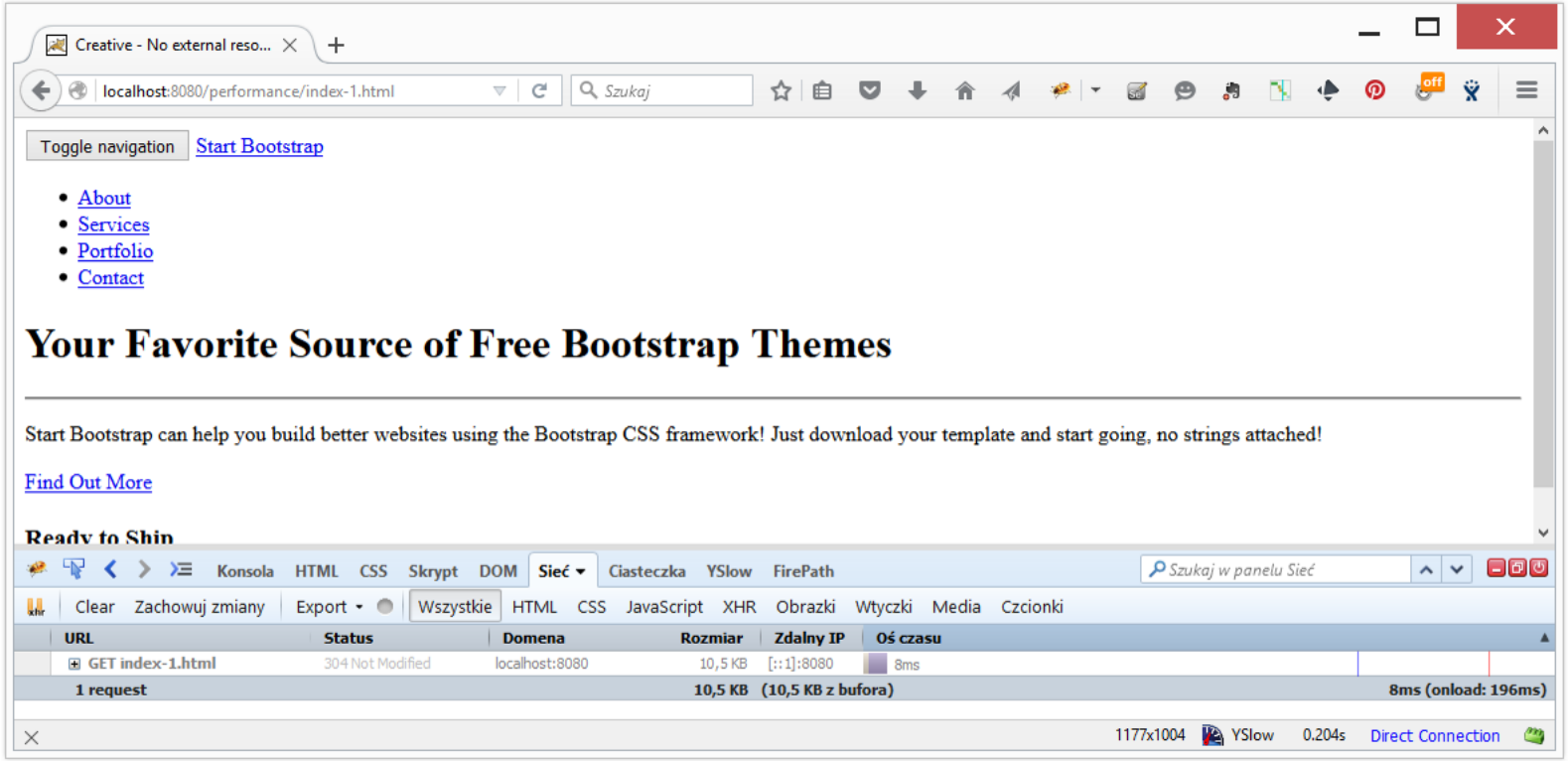
Performance testing with load generators

- Overview
  - Performed with tools like Jmeter, Gatling, AB, LoadRunner etc.
  - Generates concurrent requests based on scripted instruction
  - Provides metrics based on response time for request

- Advantages
  - Requires relatively small amount of hardware for simulating huge amount of concurrent users
  - Loads server resources (CPU, memory) and network
  - Easy to run and collect results

- Disadvantages
  - Does not work exactly as browsers

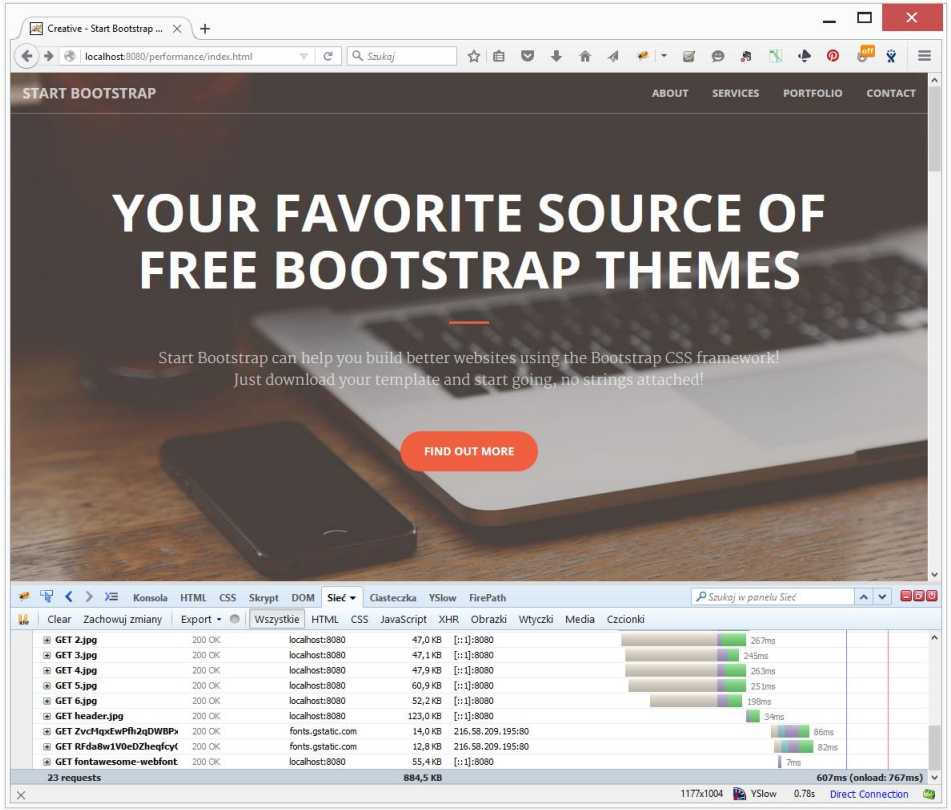
Browser vs load generator results



USING REAL BROWSER IN PERFORMANCE TESTS

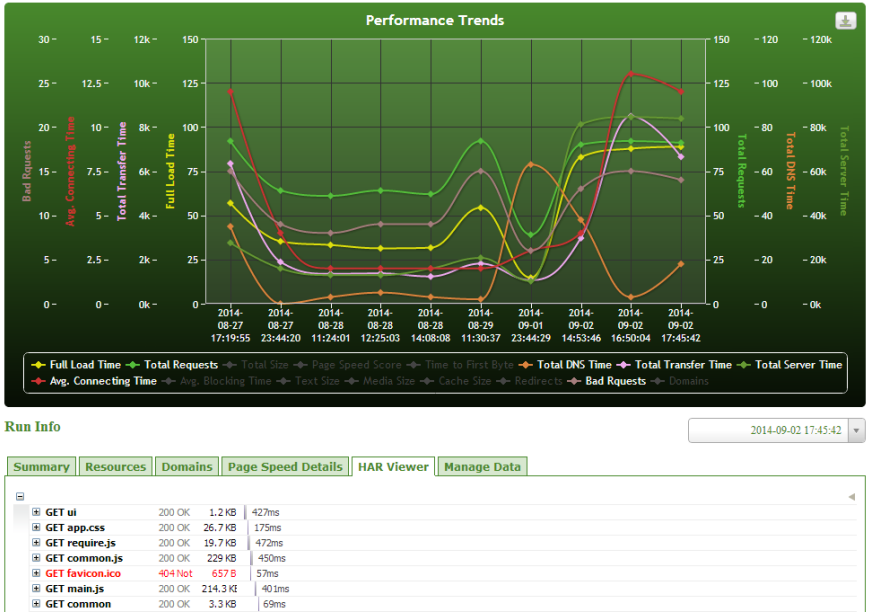
- Key advantages
  - Provides metrics from real browser
  - You can run them when you want
- Key disadvantage
  - User perspective as if she use test equipment

  browser, hardware, connections speed, location)
- Key challenges
  - Where to put test equipment
  - What metrics should be collected



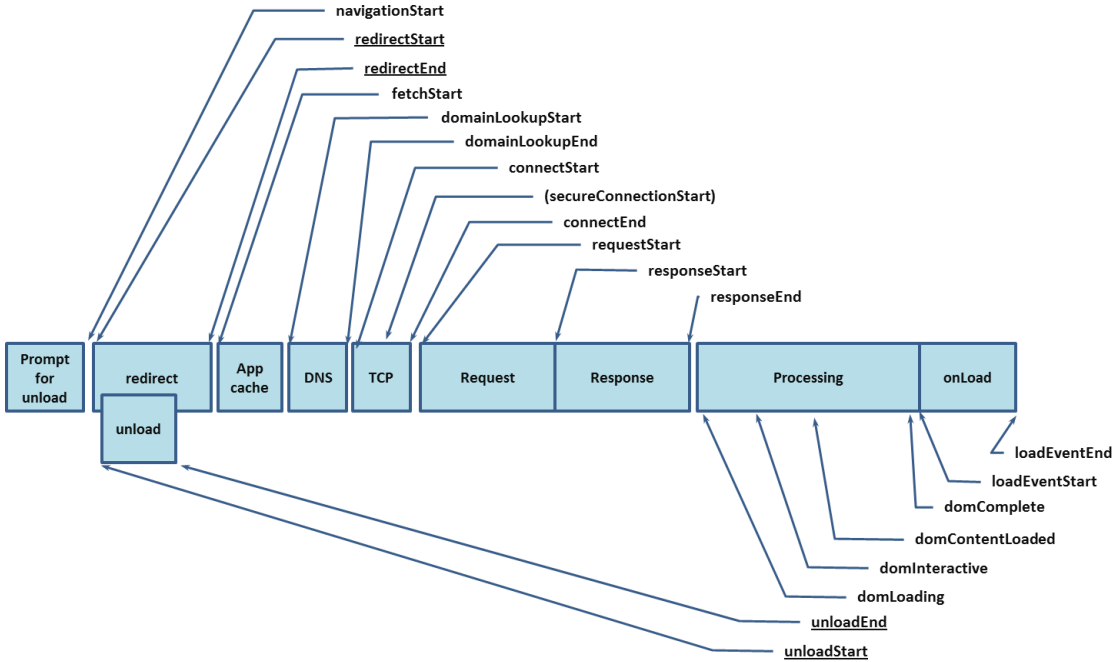
Tools and services

- Open Source
  - Selenium with BrowserMob and HarStorage – simple on site solution
  - SiteSpeed.io – tool that can crawl part or entire web site and collect yslow and performance measurements
  - WebPageTest – collect measurements from various locations, browsers and connection seeds. It captures filmstrips, allows for scripting, heavy customization and calling via API
- Commercial services
  - NewRelic – affordable tool that can be used to synthetic measurements from various locations (AWS) and RUM
  - CatchPoint – gives you give insight into application performance from lots of locations around the world



Browser events

- Common metrics
  - domContentLoaded
  - load (loadEventEnd)

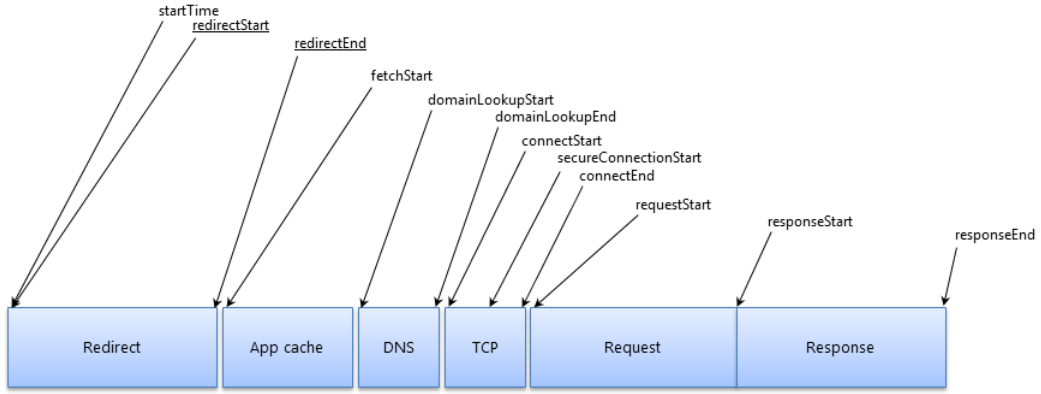


Navigation Timing API

```
var performance = window.performance ||
    window.webkitPerformance || window.mozPerformance ||
    window.msPerformance || {};
var t = performance.timing || {};
var interactive = t.domInteractive - t.navigationStart,
    dcl = t.domContentLoadedEventStart - t.navigationStart,
    complete = t.domComplete - t.navigationStart;
return interactive + ', ' + dcl + ', ' + complete;
```

Resources

- Common metrics
  - Specific resource loaded
  - Last resource loaded



Custom measurements

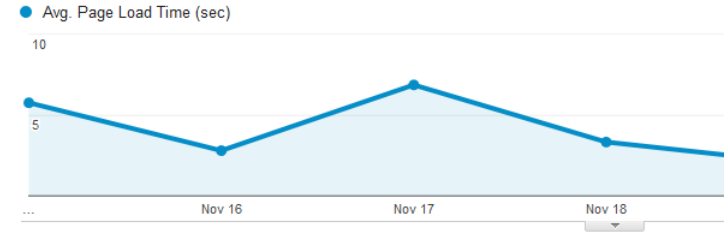
- Common metric
  - Specific resource displayed

```
<link rel="stylesheet" type="text/css"
  media="all"
  href="contentSearchUI.css">
  …
<img src="hero.jpg"
  onload="performance.mark('hero1')">
<script>performance.mark('hero2')</script>
```

* https://www.youtube.com/watch?v=f5_iAzS3WMQ

Real User Measurements (RUM)

- Collected from real user devices (mobile and desktop) and helps understands how performance is seen by real users
- Key advantages
  - You know what performance is seen by real users
  - Solves most of the challenges of synthetic monitoring
- Key disadvantage
  - You can not run it when you want (for example before release)
- Key challenges
  - What to measure



Primary Dimension: **Page** Other

Plot Rows | Secondary dimension: Country | Sort Type: Default

	Page	Country	Avg. Page Load Time (sec)
			3.42 Avg for View: 2.95 (16.09%)
1.	/	Japan	10.10
2.	/	Israel	5.75
3.	/	Belgium	5.14
4.	/	Switzerland	4.65
5.	/	Germany	3.63
6.	/	United Kingdom	2.69
7.	/	Gibraltar	2.67

References

- https://www.soasta.com/blog/measuring-web-performance-video/
- https://dzone.com/articles/a-brief-history-of-web-performance-roi
- https://developers.google.com/web/fundamentals/performance
- http://yslow.org/
- http://www.softwareishard.com/blog/har-viewer/
- https://www.soasta.com/blog/ebook-usertiming-performance-monitoring/
- https://codeascraft.com/
- https://www.soasta.com/blog/ebook-usertiming-performance-monitoring/
- https://code.facebook.com/posts/991252547593574/the-technology-behind-preview-photos/
- https://blog.twitter.com/2012/improving-performance-on-twittercom
- http://www.stevesouders.com/blog/2015/05/12/hero-image-custom-metrics

Steve Souders
@Souders

Ilya Grigorik
@igrigorik

Tammy Everts
@tameverts

Thank you

GFT Technologies SE
Jacek Okrojek
Email: jacek.okrojek@gft.com