Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Master's Thesis

Performance-optimized A/B-Testing for E-Commerce-Websites

Aram Yesildeniz

aram.yesildeniz@studium.uni-hamburg.de
M. Sc. Informatics
Matr.-No. 6890370

1st supervisor: Dr. Wolfram Wingerath
2nd supervisor: Benjamin Wollmer

Hamburg, 1st of September 2021

A distributed system is one where the failure of some
computer I've never heard of can keep me from getting my work done.
– *Leslie Lamport*

Contents

1 Introduction and Background

[tbd]

- Explain structure and main goal of this thesis
- Describe shortly all sections from this chapter and what the reader can expect
- Give short outlook to following chapter

1.1 E-Commerce

This chapter gives an introduction to e-commerce. Before going into the details, the emergence of e-commerce is described, starting with the Internet in section 1.1.1. In the following, I will briefly discuss the history of e-commerce and its types in section 1.1.2. The relationship between user satisfaction and the website performance is covered in section 1.1.3, which then leads to the chapter 1.2 which is about web analytics.

1.1.1 The Internet

In the last 50 years, a new technology emerged, spread over the entire world and influenced many aspects of most peoples life. Within the turmoil of the cold war, the United State's *Advanced Research Projects Agency* (ARPA) established in 1957 a communication network to bring together universities and their researches all around the country in order to be able compete against the USSR [CA11]. What started as a tool for scientific collaboration evolved half a century later into the *Internet*, a global network and phenomenon, to which every user with a dedicated device has access and can contribute to. The internet is an integral part, if not the backbone of today's everyday life. Users of the internet use it for almost everything, from sending emails, watching television, chatting with friends, order lunch, checking the weather for the next day or renting motorized scooters.

In 2021, the internet has 4.66 billion users, which is around 60% of the world population.[1] Compared to 2020, the number of internet users increased by 7.3%. In Europe, more than 90% of the population are internet users. For a developed country like Germany, the numbers are even more impressing: 94% of the German population are using the internet with an average daily time of over five hours.

[1] Following statistics are taken from `https://datareportal.com/reports/digital-2021-germany` [14.05.2021]

Those numbers demonstrate impressive that the internet is an integral part of our daily life. Along the rise of the internet, transactions and processes falling under the term of e-commerce are climbing as well. Before discussing the term "e-commerce" and take a grasp at its history and types, some statistics are presented to demonstrate the importance of e-commerce.

1.1.2 E-Commerce

Introduction

From the global data report[2], one can read out that over 90% of the world population visited an online retail site and over 76% of the world population purchased a product online. As usual, for or a western country like Germany, the figures are higher: 92.5% of the German population visited an online retail site and over 80% purchased a product online. And the usage is expanding: the growth of the amount spent within the category food and personal care is 28.6%, and 17.6% for the category fashion and beauty.

E-commerce sales have grown steadily over the past 20 years, topping to 57.8 billion in 2019.[3]

The COVID-19 pandemic with its implications had and still has an not negligible impact on the growth of e-commerce. Several measures were taken to stop the spread of the virus and the number of deaths, one of which was to minimize physical interaction between people. This leads consequently to a shift of human interactions to the internet. Along this, e-commerce benefits. Bhatti et al. [BAB+20] conclude that "e-commerce enhanced by COVID-19".

Brief History

E-Commerce, or electronic commerce, is according to the *Encyclopædia Britannica* about "maintaining relationships and conducting business transactions that include selling information, services, and goods by means of computer telecommunications networks."[4] In short, e-commerce is about buying and selling products and services via the internet.

The success of e-commerce is closely linked to the tremendous advances in Internet technology in recent years: The development of the *Electronic Data Interchange* (EDI) starting in the 1960s standardised the communication between two machines. Personal computers were introduced in the 1980s, and one of the first examples of an online shop is the *Electronic Mall* opened by CompuServe in 1984. Another crucial milestone is the launch of the *World Wide Web* (WWW) in 1990, which made the internet accessible to everyone. With social media visible on the horizon from the 2000s, new possibilities for

[2] https://datareportal.com/reports/digital-2021-germany [14.05.2021]
[3] https://einzelhandel.de/presse/zahlenfaktengrafiken/861-online-handel/1889-e-commerce-umsaetze [14.05.2021]
[4] https://www.britannica.com/technology/e-commerce [19.05.2021]

businesses and consumers alike to participate in e-commerce arise, for example, by enabling new marketing strategies or providing new sales channels. New devices such as smart phones and tablets lowered the barrier to participate in e-commerce. While e-commerce was available at any time, the new devices brought flexibility and mobility, making e-commerce available everywhere [Her19].

With the continued advancement in technology, e-commerce can expect a bright future with trends such as AI recommendation systems, outstanding UX thanks to virtual reality, or even more simpler payment methods through cryptocurrencies.[5]

Types

There are several types in e-commerce and they emerge from the possible combinations between the actors *business*, *consumer* and *government* [SSMG17].

	Business	Consumer	Government
Business	B2B	B2C	B2G
Consumer		C2C	C2G
Government			G2G

Table 1.1: Types of e-commerce.

B2C Business to Consumer in e-commerce describes basically online shopping, by means of a business offering its services and products to the consumer over the *WWW*. The consumer can browse through the products and services presented within an online shop and order them directly via the website. A variety of payment and delivery options conclude the B2C type [Hei20].

For an aspiring business, there are several ready-made software solutions for setting up an online store, as for example. *Shopify*, *ePages*, *Magento* or *WooCommerce* [SBR+19].

A famous example of a B2C company is *Amazon*. On the 16th of July in 1995, Amazon launched as a website and entered the stock market on the 15th of May 1997 [SB19]. Amazon has been successful, with the stock starting at $1.5, which is at around $3200 as of this writing.[6] Today, Amazon employs over 1 million people[7] and serves the desires of 200 million paying prime members.[8]

By taking a quick look at the pros and cons of an online store, it becomes clear that some of the advantages are that: there is no need of a real, physical store to showcase and sell the products; the virtual shop is available to the consumer at any time and has no closing

[5] https://www.spiralytics.com/blog/past-present-future-ecommerce/ [19.05.2021]
[6] https://finance.yahoo.com/quote/AMZN?p=AMZN [19.05.2021]
[7] https://www.statista.com/statistics/234488/number-of-amazon-employees/ [19.05.2021]
[8] https://www.statista.com/statistics/829113/number-of-paying-amazon-prime-members/ [20.05.2021]

hours; there is a high potential for the online shop as it is part of growing market; online business is scalable; due to tracking algorithms, precise targeting as well as data analysis is possible; to start an online business, there is not so much floating required and there are generally lower costs; it is possible to provide a personalized customer experience.

Some disadvantages are that the speed of market is rapid, competitors arise everyday everywhere and technology evolves quickly while consumers expectations go high [Her19], [LO20].

Another disadvantage is that there is no direct or physical connection with the consumer. As described above, online shopping takes place on the virtual WWW, i.e. personal interaction between buyer and seller is not possible and the shopping experience takes place on a website, from which it follows that the overall virtual user experience must be excellent in order to compete.

In the next section, I will describe the findings between the correlation between user satisfaction and the performance of the retailers web presence.

1.1.3 User Satisfaction and Performance

The aim of this thesis is not to deep dive into terms and concepts or the non-trivial problem of defining user satisfaction, usability or the like. Therefore the term user satisfaction is in this context loosely defined as how happy the user is with the website he or she interacts with.[9]

Performance can be understood as the speed of an online shop, e.g. how long it takes the page to load, how quickly the user can interact with the page, and how the user perceives the performance of the website. In chapter X I will discuss that measuring performance is not so trivial and there are a lot of ideas and metrics to measure it.

SpeedHub

A plethora of information and studies about the phenomenon of user satisfaction and web site performance is collected at *SpeedHub.org*, a portal by *Baqend* in cooperation with *Google* which provides "the largest systematic study of Mobile Site Speed and the Impact on E-Commerce."[10] Not only are studies and reports available on the hub, but also collections of videos and blog posts.

In his talk at code.talks 2019, Felix Gessert summarizes the results and provides insights into the most important aspects and questions of the study so far [Ges21]:

The first observation when asking for a correlation between the performance of a system and user satisfaction is that users need to be differentiated, which leads to the concept of a *User Profile*: In terms of gender, young women are the most demanding consumers

[9] For a discussion cf. "User satisfaction measurement" in [IKOK10]
[10] `https://www.speedhub.org/` [21.05.2021]

and are less likely to buy from slow sites. In general, people between the ages of 18 and 24 have higher expectations of a site's speed than their older counterparts.

There are also differences between nations and regions, for example people from Japan have the highest expectations, which is almost certainly related to technological advancements in that country. Not only the expectations themselves differ geographically, but also how speed influences the users, for example "speed influences New Yorkers more than Californians." [Unb], [GA]

What all users have in common is their human psychology. In terms of performance, researchers generally suggest keeping wait times below one second to keep users' attention. (cf. "Performance perception" in 1.2.5).

After considering the user himself, the next step is to examine the influence of the device used: Studies show that mobile users are more likely to buy products and services than their colleagues using a desktop computer, where iOS users have generally more expectations regarding site speed [Dev].

Last but not least, the context and state of the user is important, with naturally relaxed and calm users perceiving pages faster than stressed or hurried users. Also users experience websites more slowly while on the go [Akab].

There are many real world examples and studies that prove and demonstrate the importance of website speed in terms of user satisfaction and ultimately sales: *Amazon* found out that a decrease of 100 ms in page loading leads to -1% conversion rates. If the site loads 100 ms faster, *Walmart* observed that the revenue increases by 1%. For *Zalando*, increasing site speed by 100 ms has led to an uplift of 0.7% revenue per session [Lin], [SKG+], [CKR].

Search engine optimization is heavily impacted by load speed: For *Google*, 500 ms slower sites led to a decrease of 20% in traffic. *GQs* traffic increased by 80% after the page load went down from 7 s to 2 s. And for *Pinterest*, 40% faster loads led to 15% more SEO traffic [Mos], [May], [MAC].

User engagement and satisfaction also depend heavily on loading times: *Forrester* noted an increase of 60% for the session length while brining down the load time by 80%. *Akamai* monitored that the bounce rate climbed up incredible 103% when the load time increased by 2 seconds. And for the *AberdeedGroup*, the customer satisfaction dropped by 16% at one more second delay in response times [For], [Akaa], [Abe].

In summary, it can be said that many studies and practical examples prove and demonstrate that faster websites and online stores lead to a better user experience and usually to happier customers. In commercial terms, one can conclude that page speed equals money.

In order to properly test the effects of performance on users, a scientific method is required. A/B testing as a controlled experiment is one of them and will be explained in

the next section. After discussing A/B testing, I will move on to examining *Web Analytics*, a term that encompasses methods, tools, and instruments for companies to better understand their business and customers.

A/B Testing

Controlled experiments like A/B testing are not a new tool for scientists and researchers and were used as early as the 1920s [KL17]. With the advent of the Internet in the 1990s, the concept was adopted into the online domain and is now used by large companies such as Amazon, Facebook or Google to test ideas and hypotheses directly on a live system. Controlled experiments such as A/B testing are used to aid decision making and provide a "causal relationship with high probability" [KL17]. They enable a data-driven and quantitative validation of the hypothesis [Mor18].

Controlled experiments help to test hypothesis and questions of form: "If I change feature X, will it help to improve the key performance indicator Y?"

To answer this question, two systems are needed: *Version A*, the control variant or default version, and a slightly different *Version B,* called the treatment. If more than two versions or one treatment should be evaluated at the same time, an A/B/n split test has to be implemented. With a univariable setup, only one variable differs between the systems; with a multivariable structure, several variables are changed at the same time.

Usually, the users of the system are randomly split into two groups and testing is directly performed with real users on a production system. It is advantageous, also compared to other experimental set-ups, that the users and participants are not aware that they are part of an experiment, which leads to fewer biases and side effects. In order to measure the differences and the user behaviour, web analytics has to be integrated within the system [KL17].

A brief and general discussion of controlled experiments in computer science can be found in chapter X.

To continue with the question of performance and user satisfaction, A/B testing allows two different versions of the same site to be served to two groups at the same time, one site being slow and the other being fast, without users knowing.

An implemented web analytics system makes it possible to measure how the various systems and user groups behave. What web analytics exactly is, what tools are available and what a web analytics process looks like, is discussed in the next section.

1.2 Web Analytics

This chapter is about web analytics. I will first discuss definitions and give a short introduction to web analytics in section 1.2.1. Then, a brief history of web analytics will be given in section 1.2.2 to help contextualize. After characterizing two web analytics

process descriptions in section 1.2.3, data collection methods will be discussed in section 1.2.4. Finally, web performance will be explained in 1.2.5, which leads the way to the research questions.

1.2.1 Introduction

What is *Web Analytics*? Reviewing the literature, it is clear that there are several definitions:

Nakatani et al. state that "Web analytics is used to understand online customers and their behaviors, design actions influential to them, and ultimately foster behaviors beneficial to the business and achieve the organization's goal." [NC11] According to this definition, web analytics is about getting insights of the users using the system, not only who or what they are, but also how they interact with the system. Additionally, the definition emphasizes that the underlying motivation of web analytics is to achieve business goals.

Singal et al. provide a more technical definition by pointing out that "Web Analytics is the objective tracking, collection, measurement, reporting and analysis of quantitative internet data to optimize websites and web marketing initiatives." [SKS14] Again, the ultimate goal is to drive business, but supported by data science methods and tools such as tracking, collecting and analysing massive amounts of data.

Bekavac et al. provide a similar definition by pointing out that web analytics is "the analysis of qualitative and quantitative data on the website in order to continuously improve the online experience of visitors, which leads to more efficient and effective realization of the company's planned goals." [BGP15]

Summarizing the above definitions, it is noticeable that web analytics consists of two important elements: a data-driven, information-oriented and technical element of collecting and analysing data about users and a commercial and business-driven element that provides the main motivation for collecting the data primarily by setting business goals.

Moving from definitions to the practical realm, Zheng et al. describe four main use cases for web analytics [ZP15]:

- Improving the overall design and usability, for example of the navigation or layout of the website.
- Optimize for your business goals: Whatever goals the business is trying to achieve, generating conversions is the goal.
- Monitor campaigns: Understand and measure the success of advertising campaigns.
- Improve performance by examining metrics such as page load time. This is discussed further in chapter 1.2.5.

Web analytics is also difficult and there are some obstacles and challenges to overcome. Kumar et al. describe some of the hurdles as follows [KO20]: The analysis and interpretation of the data and the goals and measures derived from it are reactive rather than anticipatory, since "predictive modelling applications" for web analytics are not yet on the market.

Big data, that is the volume, variety and velocity of data collected, can be challenging to manage, for example, important insights can be lost or overlooked.

Privacy and the information collected from visitors and customers can be another challenge in terms of inappropriate use of data and GDPR compliance.

1.2.2 Brief History

The history of web analytics can be described as a transformation from an IT domain and a technical log file analysis tool to a sophisticated, polymorphic tool for marketers.

Each time a user requests an HTML file or other resource from a web server, the server makes an entry in a special log file [SKS14]. The first log entries followed the *Common Log Format* (CLF) which provides rudimentary information such as the date, the HTTP status code or the number of bytes transmitted. In 1996, the *Extended Log Format* (ELF) was introduced with more flexibility and information in mind. Thanks to the standardized format of the log files, it was possible to create software that evaluates the log files and presents them to users in a readable form. *GetStats* was one of the first tools which generated statistics and user friendly output for analysts [CP09]. In 1995, Dr. Stephen Turner developed *Analog*, the first free software for analysing log files [ZP15]. Log file analysis will be further discussed in chapter 1.2.4.

What was initially mainly interesting for maintenance and IT staff, who for example answered the question of how many 404s occurred on the server, developed into an interesting website information pool for marketers.

As the available information increased, it became clear that the data could be used for more than just analyzing server behavior. But log file analysis was not enough to provide details about how users interact with the site. Web analytics underwent a transformation from log analysis to user data tracking, analysis, and reporting.

Croll describes the move of analytics from IT to marketing with three steps: [CP09]

1. JavaScript eliminated the need for log files and enabled marketers to maintain and deploy their analytics solutions themselves, making them less dependent on IT.

2. The introduced advertising economy of search engines like Google led to a new focus of analysts on user attraction and conversion rates.

3. New cost models allowed marketers to pay for the analytics service based on website traffic, rather than paying upfront for hardware and software. Analytics spend was thus linked to website traffic and, ideally, revenue.

In 1990 the WWW started and in 1993 one of the first widely used browsers *Mosaic* was launched. At the same time *WebTrends* developed and released one of the first analytics software. A lot more services followed, such as *WebSideStory* in 1996 [CP09] or *Quantified* by Urchin in the same year [CP09].

Page tagging made it possible to collect not only technical data, but also business-relevant information. Visitors and their behaviour were the focus, shaping questions like: How is this user behaviour related to a purchase? If a user buys shoes, will he also buy socks? The development and implementation of cookies enabled the identification of unique users. Not only business-relevant questions were asked and answered, but also studies on performance and usability [CP09]. More details about page tagging can be found in chapter 1.2.4.

In 2003, Edwards, Eisenberg and Sterne founded the *Web Analytics Association* (WAA). The WAA brings together and supports all the players in web analytics such as users, marketers and IT specialists on an international stage. Due to digitalization and its all-encompassing effects, the WAA has renamed itself *Digital Analytics Association* (DAA) in 2012 because the web is not the only area where users leave their digital footprint [SKS14].

As described in the section above, participant and user numbers on the Internet continue to rise and now all Fortune 500 companies operate websites, with web analytics a key marketing tool [KO20].

Some of the most established tools today are *Google Analytics*, *Adobe SiteCatalyst*, *Webtrekk* and *Piwik* [Hei20]. Google Analytics will be further discussed in chapter **??**.

Looking ahead, Zheng et al. identify several trends for web analytics, such as mobile web and application-specific analytics like video, search, learning, or social media analytics [ZP15].

1.2.3 Web Analytics Process

Web analytics can be described as a process in which the main goal is usually to increase sales. The literature cites two main ideas and processes, the first from the Web Analytics Association and the second from industry best practices. They are briefly described in this section.

Both processes have in common that they are aimed at improving the website and thus increasing business revenue.

Key Performance Indicators (KPIs) are the ideal tool for the instrumentation of web analytics and help to identify areas and potential for improvement. They are an integral part and play an important role in any web analytics process as they provide a "in-depth picture of visitor behavior on a site" [Jan09].

KPIs can differ depending on the business in which they operate. For commercial domains, common KPIs are conversion rates, average order or visit value, customer loyalty, bounce rate, etc. [SKS14].

Defining the right KPIs and aligning them with business goals is a critical step in any web analytics process.

WAA Process Guide

The Web Analytics Association offers a web analytics guide that consists of nine steps. They are: [Jan09]

1. Identify key stake holders
2. Define primary goals of website and prioritize them
3. Identify most important site visitors
4. Determine key performance indicators
5. Identify and implement the right solution
6. Use multiple technologies and methods
7. Make improvements iteratively
8. Hire and empower a full-time analyst
9. Establish a process of continuous improvement

Industries Best Practices

On the contrary, Waisberg and Kaushik derive a five-step process from industry best practices with the main goal of improving the website and increasing sales [WK09]:

- Define Goals
- Build KPIs
- Collect Data
- Analyse Data
- Implement Changes
- Repeat last two steps

When comparing the two proposed processes, it becomes clear that both focus on identifying and being aware of the most important business goals. The WAA process is a finer-grained and more practical approach, with Waisberg and Kaushik abstracting the main activities.

1.2.4 Log File Analysis and Page Tagging

There are four main methods of collecting data for web analytics: through log file analysis, JavaScript page tagging, web beacons, and packet sniffing [WK09].

As already mentioned in chapter 1.2.2, the two most important methods of data collection are log file analysis and page tagging. In this section I will briefly describe and compare both mechanisms.

Log File Analysis

As already described, the log file analysis is about gaining knowledge from the log file records of the web server. Log file analysis is considered to be the traditional and original approach to web analytics [Mar15], [ZP15]. As soon as the user types a URL into the browser and presses the enter key, the request arrives at a web server. The server then creates an entry in a log file and sends the requested page or resource back to the client in response [WK09]. The information within the log entry can vary depending on the log format, usually IP, browser, time stamp, time required, transferred bytes, whether a cache hit occurs and the referrer is specified [WK09]. The standardized Common Log Format provides host, ident, authuser, date, request, status, bytes, as can be seen in listing 1.1.[11]

Listing 1.1: CLF

```
127.0.0.1 user-identifier frank [10/Oct/2000:13:55:36 -0700] "GET /apache_pb.gif HTTP/1.0" 200 2326
```

Standard formats of log files and entries enable log file analysis software to process, evaluate and report valuable statistics such as *Analog*, *Webalizer* or *AWStats* to users [ZP15].

Below are a few points that describe the pros and cons of log file analysis. The advantages of log file analysis are ([WK09], [NC11], [SKS14], [ZP15]):

- JavaScript and cookies are not required on the client side
- Maintainer of the website and server owns the data
- Bots and web crawler requests are also logged
- History of data is available
- Log entries are reliable
- The standard format of log files enables easy switching of analysis tools
- The web server also logs failed requests

[11] https://www.w3.org/Daemon/User/Config/Logging.html#common-logfile-format [03.06.2021]

- No modification on the web page needed
- Does not demand more bandwidth

Some disadvantages of log file analysis are ([Mar15], [ZP15]):

- Log entries provide mainly technical information, which may not be very useful for user behaviour analysis. Business related metrics such as bounce rates are not available.
- Only direct requests from the client to the web server are logged: Any user interaction in the web browser that does not trigger a request is not logged. Responses from caches and proxies are also not visible in the web server's log file. Only interactions with the web server are logged.

Page Tagging

In a nutshell, page tagging describes the analysis method used to incorporate JavaScript into the website, which collects data and sends it to an analysis server [Mar15]. To do this, JavaScript must be integrated on every page to be analysed [WK09]. Page tagging is the most important method in web analytics today [ZP15].

Croll provides a illustration (figure 1.1) of the page tagging process which is explained below [CP09].

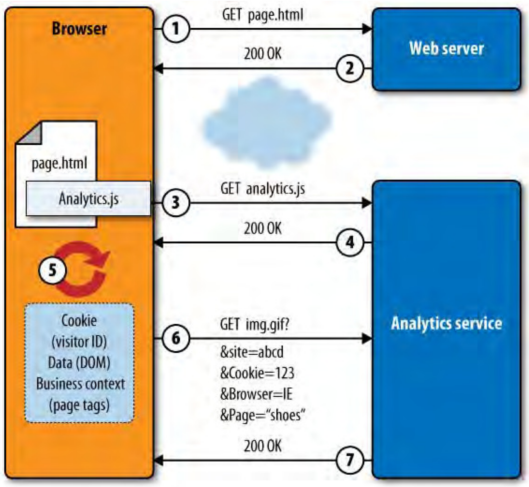


Figure 1.1: Page Tagging

The client (browser) requests a page from the web server (1, 2). Within the HTML file an external JavaScript resource, the analytics code, is linked and received from the analytics server (3, 4). The analytics script tracks and measures the user behaviour and eventually sends the data back to the analytics server (5, 6, 7).

The data collected can also be stored in cookies, which contain data beyond a session and enable the user to be identified, e.g. the next time he visits the page [KO20].

The advantages and disadvantages of page tagging are as follows, starting with the pros ([WK09], [NC11], [Mar15], [SKS14], [ZP15]):

- Every page visit is counted
- The analytics service is outsourced, which includes the storage of the data, but also the data analysis and reporting
- Page tagging is rather easy to implement and favourable when the analyst does not have access to the web server
- Highly customizable: Everything that JavaScript enables to measure, collect, and track is available. This also includes information about the client such as screen size, device used or color depth.
- Ability to track events and actions such as mouse clicks that do not send requests to the web server. This is especially important for single-page or progressive web applications that do not generate requests as often.
- Mechanics of cookies provide identification of unique and repeat visitors
- Real time reporting is possible

Some drawbacks are mainly privacy concerns, that the analysis process relies on the use of JavaScript and cookies that can be disabled by the user [Mar15], that every page that is supposed to collect data must contain the analytics script and due to the use of a third party analytics service it is pretty difficult to switch tools [SKS14].

1.2.5 Web Performance

This chapter gives only a brief overview of the various aspects of web performance. Reference is made below to the appropriate sections that cover the topics in more detail.

As already described in chapter 1.1.3, web performance plays a role that cannot be neglected for user satisfaction and business success. The above studies show that increasing website performance also increases sales, or as Google states it: "Performance plays a major role in the success of any online venture".[12]

The MDN Web Docs identify multiple areas of web performance:[13]

- Reducing overall load time
- Making the site usable as soon as possible

[12] `https://web.dev/why-speed-matters/` [03.06.2021]

[13] `https://developer.mozilla.org/en-US/docs/Learn/Performance/What_is_web_performance` [03.06.2021]

- Smoothness and interactivity
- Perceived performance
- Performance measurements

Reducing load time The question of what makes websites slow is covered in chapter X.

Usability and interactivity As I will describe in chapter X, there are several metrics available that attempt to reflect areas of performance such as load time, smoothness, and interactivity, and specific metrics are available as well as for differentiating between technical and user-perceived performance.

Performance perception The perception of performance is generally subjective. As already seen in chapter 1.1.3, there are some quantifiable time intervals that correlate with human psychology regarding the received performance. Table 1.2.5 contains "unofficial rule of thumb" for delay thresholds [Gri13].

Delay	User Perception
0-100 ms	Instant
100-300 ms	Small perceptible delay
300-1000 ms	Machine is working
> 1 s	Likely mental context switch
> 10 s	Task is abandoned

Table 1.2: Rule of thumbs for delay

If one interprets the numbers from the table, one can make the statement that it is desirable to keep loading times below one second. Thresholds for certain performance metrics and the psychological rationale for setting them are discussed in chapter X.

Performance measurements There are several methods of measuring performance. *Synthetic monitoring* is discussed in chapter X. *Real User Monitoring* (RUM) is covered in chapter X.

1.3 Research Question

The e-commerce industry is booming and there are no signs that this trend is reversing; on the other hand. Performance plays an important role in terms of customer satisfaction and how this directly affects business revenue. To better understand e-commerce website visitors, page tagging is widely used and implemented.

Several questions and issues can arise in this area and context, such as: Does page tagging affect the website's performance? Intuitively, it can be said that loading additional

JavaScript will reduce the performance of the website, depending on parameters such as the script size and network condition. But are there more unpredictable side effects? Do the various techniques of embedding a tracking script affect the data collected and measured? Will the various tracking scripts supplied interfere with each other?

A hypothesis of this work is that tracking tools slow down the monitored websites, reduce the speed and performance of the website and thus have an unfavourable effect on the user experience.

These questions are to be investigated within the scope of this thesis.

1.3.1 Goal

This thesis has several goals:

The Internet and websites in general are complicated, complex, and tangled. Although basic HTML structures are standardized, each website follows its own form and is unique and sui generis. In order to conduct a controlled experiment and test hypotheses, one goal is to approximate real websites with an artificial, laboratory-generated website that is completely controlled and manipulated by the researcher.

The aim is to create a reliable, but also convincing test environment in order to model and reproduce real behaviour.

Once the test environment is up and running, performance measurement issues need to be addressed. The aim is to measure, collect, visualize and analyse performance data.

As we will see in chapter X, there are many metrics for measuring performance. Another goal of this work is to establish something like a taxonomy of performance metrics.

1.3.2 Chapter Outline

[tbd]

Chapter 1 was about... In Chapter 2 we see, Chatper 3...

2 Metrics and Measurement Methods

[tbd]

- Last chapter...
- This chapter: Describe shortly all sections from this chapter
- In the next chapter...

In this chapter, I will cover measurement methods and discuss common performance metrics.

- This chapter should cover all relevant terms and definitions within web performance measurement
- How terms can be structured / taxonomy
- Ambiguity of definitions

2.1 Why are Websites slow?

Possible sections:
- Page Speed
- Critical Rendering Path: How the Browser works, what exactly happens with the received html etc. - Part of this ? or even before is the connection with this famous image
- Regarding Networking the Latency issue is important
- Talk a little bit about FE improvement ?
- For details on networking like TCP TLS etc always cf to Grigorik

2.1.1 Navigation

-> This is basically a description of this famous image regarding navigation
Overview ———
- Latency as main threat - "Network latency is the time it takes to transmit bytes over-the-air to computers" - Browser is single threaded - Enable smooth interaction: scrolling, responsive to touch, etc. - Render time is key - Goal: Main thread can complete all the work and still is available to handle user interaction -> Improvement: Understand single

thread concept of browser and minimize main threads responsibilities -> Should lead to: rendering is fast and smooth and responses to interactions are immediate

Navigation ———-

- First step in loading a web page - Goal: minimize the amount of time a navigation takes to complete

DNS Lookup: - translate URL to IP address. Can be cached by browser - Must be done for each unique URL, e.g. when images are from different server

TCP Handshake: - 3-way-handshake between client and server -> 3 more requests between client and server

TLS Negotiation: - Make connection secure by changing cipher etc -> 3 more round trips

-> At this point already 8 round trips

Request/Response ——————-

- HTTP Get request by client to server - Response contains first byte of data received -> Time to First Byte Metric! -> link here to metric description

TCP Slow Start / 14kb rule: - Is an algorithm: Slow start gradually increases the amount of data transmitted until the network's maximum bandwidth can be determined - Avoid congestion - This is not part of TCP handshake but for first HTTP request - Initial response is 14Kb, second one is 28, etc. - Congestion Control algorithms: determine send rate

Parsing ——-

- Starts when browser received fist data - HTML, CSS, and JavaScript have to be parsed. - Parsing is the step the browser takes to turn the data it receives over the network into the DOM and CSSOM, which is used by the renderer to paint a page to the screen. - DOM: internal representation of the markup for the browser - DOM is also exposed, and can be manipulated through various APIs in JavaScript - Browser will begin parsing as soon as data received: Important to include most important stuff in first 14Kb

- Build CRP: Here 5 steps from CRP are described: 1. Building DOM tree 2. Building CSSOM 3. Style 4. Layout 5. Paint

1. Building the DOM tree: - Build DOM tree by processing HTML markup (tokenization, tree construction) - Browser requests non-blocking resources and continues parsing - CSS does not block parsing - scripts which are not async or defer block parsing -> excessive scripts can be a significant bottleneck

Optimisation: Preload scanner: - This process occupies main thread while browser is building DOM Tree - parse through the content available and request high priority resources like CSS, JavaScript, and web fonts. - will retrieve resources in the background so that by the time the main HTML parser reaches requested assets, they may possibly already be in flight, or have been downloaded - CSS fetch does not block HTML parsing but JS

2. Building the CSSOM: - Process CSS and build CSSOM tree - Similar to DOM - Usu-

ally very fast

Other Processes: - JavaScript Compilation: JavaScript is interpreted, compiled, parsed and executed. - Building the Accessibility Tree (AOM)

Render ——-

3. Style: - Build Render Tree - Create render tree out of DOM and CSSOM

4. Layout: - Run layout on render tree - Compute geometry of each node - Exact size and location of each object - Subsequent changes are called reflow, e.g. when image is received later without specify its size

5. Paint: - Paint nodes on screen -> First Meaningful Paint

Interactivity ————

- Time to Interactive

- time it takes for a packet of data to travel from source to a destination

What is Latency?: - amount of time it takes from when a request is made by the user to the time it takes for the response to get back to that user - For first request latency is longer as it includes a DNS lookup, a TCP handshake, the secure TLS negotiation - Latency describes the amount of delay on a network or Internet connection - One of the main aims of improving performance is to reduce latency. - We can determine the amount of latency by measuring the speed with which the data moves from one network location to another. - Latency can be measured one way or entire round-trip - Generally measured in round-trip

Network throttling: - Emulate download speed, upload speed, and minimum latency

Network Timings: - Blocked: When a request is in queue - Blocking happens when there are too many simultaneous connections to single server over HTTP - DNS resolution: Time it took for DNS lookup - TCP, TLS - Sending, Waiting, Receiving

Performance APIs: - Many Web APIs available - Performance API is one API within Web APIs collection: Includes other APIs - Navigation Timing API: Famous image: Exposes metrics related to navigation events

Tools and metrics: - 2 categories: Tools for measuring (reporting) and tools for improving - Reporting: - PageSpeedInsights - WebPageTest - Network: - Network Panel in DevTools

In his code talk 2016, Witt identifies three main areas or bottlenecks where bad performance is being produced: In the Frontend, the Backend, and on the network layer. - the three bottlenecks: Frontend, network, backend

FE: - crp: render and parser blocking - tools available: - Profiling: GTMetrix, WebPageTest, PageSpeed Insigths - Inlining and Optimization: Critical, PostCSS, processhtml - Minification and Compression: Goole Closure, tinyPng, Uglifycss and cssmin

Network: - DNS lookup - Initial connection: TCP Handshake and TLS handshake - time to first byte: When first page data byte receives on client side - content download -> Max 6 parallel connections

- Avg total requests

- latency impact: 2 times bandwith makes no difference, half of latency makes half of load time

- Possible Improvements: - HTTP2 - Avoid redirects - Caching headers - CDNs - Single Page Apps

BE: - load balancer, server and DB

https://developers.google.com/web/fundamentals/performance/critical-rendering-path/analyzing-crp

Chapter 1 Latency and Bandwith

- Latency as a Performance Bottleneck - Show this very famous image with 2x bandwith...

Chapter 10 Web Performance

maybe ch 2 and 6

Chapter 2 The Basics of Page Speed ————————————————-

...

2.2 Measurement Methods

- synthetic monitoring - RUM - other methods briefly described

- Explanation and comparison of synthetic and real-user monitoring with concrete examples
- Short overview of other measuring methods such as log analysis or surveys

synthetic: - lab environment: geography, network, device, browser, etc. - control variables to identify performance issues. this does not reflect real world and real user experience - automated - simulate user paths - traffic is generated artificial and is not by real users - can also be used for live system monitoring - fairly easy to implement, inexpensive

RUM: - measure from real users machine - part of page tagging (same technique with including some JS) - measures actual use cases

- Synthetic and Real-User Performance Measurement

- Log file - Synthetic: will be discussed in chapter X - RUM which will be covered in chapter X - CrUX - Surveys

- Pingdom - GTMetrix - Website Grader - Site Speed checker

2.2.1 Synthetic Monitoring

- What is it
- How does it work
- Application, real life scenario
- Examples:

  - WebPageTest
  - Google Lighthouse
  - Other solutions

2.2.2 Real-User Monitoring

- What is it
- How does it work
- Application, real life scenario
- Examples:
  - Google Analyitcs
  - CrUX
  - SpeedKit
  - Other solutions

2.2.3 Other methods

Reports such as CruX or http archive surveys log files
- Log file - Synthetic: will be discussed in chapter X - RUM which will be covered in chapter X - CrUX - Surveys

2.3 Metrics

- Compare to competitors - Compare different versions of your app - Metrics should be relevant to your users, site, and business goals - should be collected and measured in a consistent manner - analyzed in a format that can be consumed and understood by non-technical stakeholders
Check out glossary: https://developer.mozilla.org/en-US/docs/Glossary
DNS resolution / DNS lookup time
Connecting / TCP Handshake time
TLS Handshake time
Waiting / Server Response Time ??
Receiving / Download Time ??
onload Event:
Time to First Byte:
First Paint:
First Contentful Paint:
First Meaningful Paint:

Largest Contentful Paint:
Speed Index:
Time to Interactive:
Performance API - Includes Performance Timeline API, the Navigation Timing API, the User Timing API, and the Resource Timing API. ??
Navigation Timing API
Performance Timeline API
User Timing API
Resource Timing API
Performance Observer API
User Timing API Navigation Timing API: Level 1 (performance.timing), Level 2 (PerformanceNavigationTiming) ? Network Information API Resource Timing API Paint Timing API High Resolution Time API Performance Timeline API Performance Observer API Long Tasks API Element Timing API Event Timing API Server Timing API

2.3.1 Introduction

- Metrics jungle, difficulty of taxonomy
- Performance vs UX

2.3.2 "Non-Performance" metrics

- User engagement: session length, bounce rate, etc.
- Business KPIs: Cart size, conversion rate, etc.
- QA metadata: Page views, JS errors, etc.
- Hit
- Click-Through
- Page View
- Visit
- Visitor / Unique Visitor
- Referrer
- Conversion Rate
- Abandonment Rate
- Attrition
- Loyalty, Frequency and Recency

- Measuring Reach: ...
- Measuring Acquisition: ...
- Measuring Conversion: ...
- Measuring Retention: ...
- Basic metrics (see table): basic metrics are meaningless
- Advanced metrics: Customer lifecycle analysis, customer behaviour analysis
- Types: Counts, Rations, KPIs
- Definitions for all terms, like Page view, unique visitor, etc.
- Importance of setting goals
- Conversion Rate
- Kennzahlen für Websites nach Typ: ROI-Ebene, Online-Shop, ...
- Conversion Rates, pages that visitors abandon most
- Click throughs
- UGC (User generated content)
- Subscriptions, Signups
- Referring URL
- Visitor Motivaton, VOC: Voice of the Customer
- Ad and campaign effectiveness
- Findability and Search Effectiveness
- Trouble Ticketing and Escalation
- Loyalty: Ratio of new to returning visitors; average time between visits; time since last login; rate of attrition or disengagement

p.15 "whether your business benefited in some way from their visits."
The percentage of visitors that your site converts to contributors, buyers, or users is the most important metric you can track -> Conversion Rate
p. 74 Page View, first useful web analytics metric

- 4 categories: site usage, referrers, site content analysis, quality assurance
- 8 fundamental metrics

- Site usage:
  - Demographics and System Statistics
  - Internal Search Information
  - Visit Length
  - Visitor Type
- Referrers:
  - Referrering URL and Keyword Analysis
- Site content analysis:
  - Top Pages
  - Visitor Path
- Quality assurance:
  - Errors

- GA basic metrics: Visits, Bounce Rate, Page views,pages per visit, avg time on site, percentage new visits

- Erfolg messen und bewerten
- Traffic:
  - Page Impression / Page View
  - Visit
  - Visitor / Unique visitor
- Bounce rate
- Conversion rate
- CTR: Click-through-rate
- Session length
- Good metrics should be: Uncomplex, Relevant, Timely, Instantly Useful
- Basic metrics: Visits, bounce rate, page views, pages/visits, avg time, % new visits
- Guidance Performance Indicator (GPI) metric
- Visit count: page view, visit, unique visitor
- Visit duration: time on page, time on site.
- Bounce rate and exit rate.

- Besucheranalyse: Wie viele Besucher?, Anzahl Besucher mit Mobilgerät, Demographische Daten (Geschlecht, Altersgruppe)
- Seitenanalyse: Was machen die Besucher im Shop?, Zielseite / Startseite: Erste Seite, die ein Besucher angeschaut hat, Ausstiegseite
- E-Commerce-Analyse: Transkations-daten aus Shop, Funnel-Analyse
- Types: Anzahl, Relations, Werte
- Content: Where, Who, How, What
- Hits
- Page Views
- Visits / Sessions
- Visitor / Unique Visitor

2.3.3 Performance Metrics

- Introduction to the Web Performance Working Group
- Overview of Browser APIs and the data they expose: High Resolution Time API, Navigation Timing API, etc.
- If possible make one deep dive into one API: What exactly gets measured? Maybe check out html standard, v8 or chromium implementation, etc.

Standards and APIs, Browser metrics, standards

- Web Performance Working Group
- User Timing API
- Navigation Timing API: Level 1 (performance.timing), Level 2 (PerformanceNavigationTiming) ?
- Network Information API
- Resource Timing API
- Paint Timing API
- High Resolution Time API
- Performance Timeline API
- Performance Observer API

- Long Tasks API
- Element Timing API
- Event Timing API
- Server Timing API

Navigation Timing API

- Show image of navigation timings
- Explain one or two events directly with specification: navigationStart, domInteractive, etc.

Google metrics? User-centric / UX / visual

Web Vitals

- Key questions: Is is usable, is it delightful, ...
- Types of metrics
- important metrics
- custom metrics
- Core Web Vitals: First Input Delay, Cumulative Layout Shift, Largest Contentful Paint
- First Paint, First Contentful Paint: Is it happening? PerformanceObserver
- First Meaningful Paint, Hero Element: Is it useful?
- Time To Interactive: Is it usable? Use Polyfill
- Long Tasks: Is it delightful? PerformanceObserver
- Total Blocking Time
- Time To First Byte

Core Web Vitals

- Most important metrics, Apply to all websites, Measures real user experience, Measurement support for Lab and Field, Concise and clear
- LCP: Progressive loading. FCP may become a core web vital
- FID: Interactivity during load

- CLS: Visual stability
- Future goals: Better support for Single Page Apps, Input responsiveness, Scrolling and animations
- Areas of user experience beyond performance: Security, Privacy, Accessibility
- Introduction, what is it
- How to measure
- How to improve
- Introduction, what is it
- How to measure
- How to improve
- Introduction, what is it
- How to measure
- How to improve

Others

- Visually complete ?

Speed Index

2.3.4 WebPageTest Metrics

- Metrics Categories:
  - High Level Metrics:
    - Document Complete
    - Fully Loaded
    - Load Time
    - First Byte
    - Start Render
    - Requests
    - Bytes In (Page Size)
  - Page-level Metrics:
    - Technical Page Metrics:

            - -> APIs, GA Site Speed Metrics
            - TTFB
            - loadTime
            - docTime
            - ...
        * Visual Metrics:
            - SpeedIndex
            - firstPaint
            - firstContentfulPaint
            - firstMeaningfulPaint
            - ...
        * Javascript and CPU timings
        * Page Information
        * Browser State
        * Lighthouse Summary Metrics
        * Optimization Checks/Grades
        * Instrumented Metrics
        * Test Information
        * Misc
    – Request-level metrics:
        * Request Details
        * Request Timings
        * Request Stats
        * Headers
        * Protocol Information
        * Javascript/CPU details
        * Optimization Checks
        * Misc

- Optimization Grades:
    – Keep-alive Enabled
    – Compress Text
    – Compress Images

  - Cache Static Content
  - Use of CDN

- First View and Repeat View

Name	Description
Successful Tests	Amount of tests who completed successfully
Document Complete	The time from the initial request until the browser fires load event. Also known as the document complete time. This is the time at which the Document Object Model (DOM) has been created and all images have been downloaded and displayed. For most traditional web pages, the load time is a suitable metric for representing how long a user must wait until the page becomes usable. This is the default performance metric on WebPageTest. Also known as Load Time (?). Around this time, the page's script is hard at work in the load-event handler firing off more requests for secondary content. The incomplete nature of this metric is why Fully Loaded was added to the table of metrics from the previous section. window.onload (?). The point where the browser onLoad event fires. The equivalent Navigation Timing event is loadEventStart. Document Complete Time: Amount of time that has elapsed from the initial page request until the browser fires the load event. This is the time at which the Document Object Model (DOM) has been created and all images have been downloaded and displayed.
Fully Loaded	The time from the initial request until WebPageTest determines that the page has finished loading content. The page might have waited for the load event to defer loading secondary content. The time it takes to load the secondary content is accounted for in the Fully Loaded Time. The time (in ms) the page took to be fully loaded — e.g., 2 seconds of no network activity after Document Complete. This will usually include any activity that is triggered by javascript after the main page loads. The point after onLoad where network activity has stopped for 2 seconds. Specific to WebPageTest and not provided by Performance API. Fully loaded waits for 2 seconds of no network activity (and no outstanding requests) after onLoad and then calls it done (only measures to the last activity, doesn't include the 2 seconds of silence in the measurement). Fully Loaded is a measure based on the network activity and is the point after onload when there was no activity for 2 seconds.

First Byte	Time until the server responds with the first byte of the response.
Start Render	Time until the browser paints content to the screen. The time for the browser to display the first pixel of content (paint) on the screen. Time until the browser paints content to the screen. WebPageTest's own metric, determined by programmatically watching for visual changes to the page. Same as First Render?
Bytes In (Doc)	Total size of the Document Complete Requests' response bodies in bytes.
Requests (Doc)	Number of HTTP requests before the load event, not including the initial request.
Load Event Start	Time in ms since navigation started until window.onload event was triggered (from W3C Navigation Timing).
Speed Index	See Speed Index
Last Visual Change	Time in ms until the last visual changed occured. Last change is a completely visual measurement and is the last point in the test when something visually changed on the screen. It could be something as simple as an animated gif or ad even that didn't really cause much CPU work but changed some pixels on the screen. It is only captured when video is recorded because it depends on the video capture to measure it.
Visually Complete	Time in ms when page was visually completed. Is measured from a video capture of the viewport loading and is the point when the visible part of the page first reached 100% "completeness" compared to the end state of the test.

Table 2.1: Your caption here

2.3.5 Google Analytics Site Speed Metrics

Show with analytics.js that it is indeed those navigation timing api calculations.

Ec = function (a)...

GA does not really provide any UX metrics! The site speed metrics are all from navigation timing api which are measurements from the browser.

GA Site Speed Metrics (description from `https://support.google.com/analytics/answer/2383341?hl=en&ref_topic=1282106`)

`https://stackoverflow.com/questions/18972615/how-do-the-metrics-of-google-anal`

Name	Description
Page Load Sample	The number of pageviews that were sampled to calculate the average page-load time.
Speed Metrics Sample	The sample set (or count) of pageviews used to calculate the averages of site speed metrics. This metric is used in all site speed average calculations, including avgDomainLookupTime, avgPageDownloadTime, avgRedirectionTime, avgServerConnectionTime, and avgServerResponseTime.
DOM Latency Metrics Sample	Sample set (or count) of pageviews used to calculate the averages for site speed DOM metrics. This metric is used to calculate ga:avgDomContentLoadedTime and ga:avgDomInteractiveTime.
Page Load Time (sec)	The average amount of time (in seconds) it takes that page to load, from initiation of the pageview (e.g., click on a page link) to load completion in the browser.
Domain Lookup Time (sec)	The average amount of time spent in DNS lookup for the page.
Page Download Time (sec)	The time to download your page.
Redirection Time (sec)	The time spent in redirection before fetching the page. If there are no redirects, the value for this metric is expected to be 0.
Server Connection Time (sec)	The time needed for the user to connect to your server.
Server Response Time (sec)	The time for your server to respond to a user request, including the network time from the user's location to your server.
Document Interactive Time (sec)	The average time (in seconds) that the browser takes to parse the document (DOMInteractive), including the network time from the user's location to your server. At this time, the user can interact with the Document Object Model even though it is not fully loaded.
Document Content Loaded Time (sec)	The average time (in seconds) that the browser takes to parse the document and execute deferred and parser-inserted scripts (DOMContentLoaded), including the network time from the user's location to your server. Parsing of the document is finished, the Document Object Model is ready, but referenced style sheets, images, and subframes may not be finished loading. This event is often the starting point for javascript framework execution, e.g., JQuery's onready() callback, etc.

2.3.6 Comparison

- We can show this with experiments
- Load test page on a specific day only once and save timings exposed by perfor-

Navigation Timing API	WPT	GA
loadEventStart - navigationStart	Document Complete, Load Event Start	pageLoadTime
domainLookupEnd - domainLookupStart	DNS lookup, dns_ms	domainLookupTime
connectEnd - connectStart	connect_ms	serverConnectionTime
responseStart - requestStart	..	serverResponseTime
responseEnd - responseStart	..	pageDownloadTime
fetchStart - navigationStart	..	redirectionTime
domInteractive - navigationStart	..	domInteractiveTime
domContentLoadedEventStart - navigationStart	domContentLoadedEventStart	domContentLoadedTime

mance.timing object (from console)

- Calculate differences corresponding to the table
- Get GA data for that day and save it
-

3 Related Work

- Last chapter...
- This chapter: Describe shortly all sections from this chapter
- In the next chapter...
- This chapter should list research which covers and explores questions relevant for this thesis, such as:
  - Metrics: New metrics, meaning of metrics, difficulties of defining metrics, etc.
  - Overview, evaluation and comparison of measurement tools and methods
  - If available: Impact of RUM on performance

3.1 WebPageTest

- Overview
- Configuration
- Private Instances

3.1.1 Overview

- What is it
- Why to use it, Who uses it, how to use it
- Waterfall and Grades
- See in performance tab for details about grades and optimization techniques

3.1.2 Configuration

- Caching, repeat view
- Traffic shaping
- e.g. capture devtools timeline

3.1.3 Private Instances

- Architecture
- AWS
- Docker localhost
- Bulk tests

3.2 Google Analytics

- Custom metrics with Google Web Vitals as example
- Show how to include GA script (analytics.js, gtag, Tag Manager, etc.)
- Show some real life examples how script code is included into page, e.g. from Amazon, Otto etc

3.2.1 The Tracking Script

- Show code example
- Explain whats going on: script tag, create script element etc.
- Maybe also show Hotjar example to see that they are similar

3.3 Research

- Research exists about topics like: ....
- Here i will provide a list of in my eyes relevant papers, summaries them and discuss why this is important for my research

3.3.1 some title for first category

2014 Singal I. - Describes history of web analytics and tools - Provides definitions and taxonomy for metrics - Describes log file vs page tagging - Describes KPIs

II. - Lit. overview for KPIs and Web Metrics - Lit. overview for "Trust" - Lit. overview for "Fuzzy" -> What are does categories?

III. - Some other literature worth mentioning

IV. - Describes 8 open challenges for researchers

2015 Bekavac - Two parts: - 1: Some general overview of web analytics, tools and metrics, KPIs etc - 2: Empirical study about employees satisfaction of used web analytics tools

1: - 9 web business models and 5 common goals - Hypothesis: Web analytics tools track and improve a user's satisfaction with web-based business models. - Web anayltics defintion. Log files vs Site Tagging - Web Analytics process - Tools: 5 categories, Process of selecting tool, Table with features of different tools - Web metrics categories, Table with business models and their KPIs

2: - Which tools are used for which purpose / Activity - Users satisfaction

3.3.2 Research about Tools

Kaushik 2007 - Provides 3 questions which help to choose web analytics tools

2011 Nakatani - Gives some arguments why web analytics is important for business - Provides different categorizations for web analytics tools - Gives pros and cons of log file analysis and page tagging - Provides tool selection method based on AHP (Analytic Hierarchy Process)

"Web analytics tools collect click-stream data, track users navigation paths, process and present the data as meaningful information. - Categorizations: 1: By 4 different data collection methods 2: SaaS vs in-house 3: mobile vs non-mobile 4: Time lag

2016 Kaur - free vs paid - real time vs long term - hosted vs in-house - data portability

- free / open source tools - proprietary tools - Service Hosted Software - GA most popular one

3.3.3 Research about Metrics

- Dont know:

4 Approach

- Last chapter...
- This chapter: Describe shortly all sections from this chapter
- In the next chapter...
- In this chapter the practical work should be documented and explained
- Elaboration of how the practical work could help answer the research question
- Discussion of real-life setup and how experiments approach it

4.1 Empirical Research Methods

- Overview of methods
- reproduceability etc.
- validity
- Justification why following approaches are conducted as controlled experiments
- Change something: Delete this item again

4.1.1 Controlled Experiment

- Short overview about controlled experiments in computer science
- Design: Show test setup image: Independent and dependent variables
- Hypothesis testing

4.1.2 Test Setup

- What is test object (website)
- What are dependent variables: Performance metrics
- What are independent variables: Specific changes in test object (see next chapter)
- Kohavi 2016: Sample size, collect right metrics, track right users, randomization unit

Variable	Values
Position	top-head, bottom-head, bottom-body
Attribute	no attribute, async, defer
Other Script	false, true

Measure effects: Dependent Variables

- Performance metrics from Lab and Field, see terms and definitions
- But also quality of RUM data. Because we could have a nice performance but RUM will be of bad quality.

Test object / HTML Template

- Depending on different approaches / Ideas (see next chapter), template looks different
- But general structure stays the same and independent variables can be defined
- Here we show different independent variables and variants

Lab and Field

- I want to collect Lab and field data for dependent variables for comparison
- This setup is a special case because lab bots (e.g. WPT) simulate at the same time real users for RUM data

4.1.3 Independent Variables within template

- IV 1 POSITION: Position of included analyitcs script. Values: top-head, bottom-head, bottom-body
- IV 2 ATTRIBUTE: Attribute of included analyitcs script: no-attribute, async, defer
- IV 3 OTHER SCRIPT: Other tracking script included
- Other IVs not included but worth mentioning

I will compare the values from one independent variable only. Therefore, when comparing the values of one independent variable, i need to set a default value for the other independent variables. The default values are:

Position: top-head Attribute: no attribute Other Script: false

Other IVs not included but worth mentioning

- More or less infinite number of independent variables
- Again the big and important fact that each website is different

Listing 4.1: Position 1

```
<!DOCTYPE html>
<html>
    <head>
        <!-- Google Analytics -->
        <script></script>
        <!-- End Google Analytics -->

        <title>
        <meta>
        <link>
        <script>
    </head>

    <body>
        ...
    </body>
</html>
```

Listing 4.2: Position 2

```
<!DOCTYPE html>
<html>
    <head>
        <title>
        <meta>
        <link>
        <script>

        <!-- Google Analytics -->
        <script></script>
        <!-- End Google Analytics -->
    </head>

    <body>
        ...
    </body>
</html>
```

Listing 4.3: Position 3

```
<!DOCTYPE html>
<html>
    <head>
        <title>
        <meta>
        <link>
        <script>

    </head>

    <body>
        ...
        <!-- Google Analytics -->
        <script></script>
        <!-- End Google Analytics -->
    </body>
</html>
```

Listing 4.4: Attribute 1

```
<!DOCTYPE html>
<html>
    <head>
        <!-- Google Analytics -->
        <script></script>
        <!-- End Google Analytics -->

        <title>
        <meta>
        <link>
        <script>
    </head>

    <body>
        ...
    </body>
</html>
```

Listing 4.5: Attribute 2

```
<!DOCTYPE html>
<html>
    <head>
        <!-- Google Analytics -->
        <script async></script>
        <!-- End Google Analytics -->

        <title>
        <meta>
        <link>
        <script>
    </head>

    <body>
        ...
    </body>
</html>
```

Listing 4.6: Attribute 3

```
<!DOCTYPE html>
<html>
    <head>
        <!-- Google Analytics -->
        <script defer></script>
        <!-- End Google Analytics -->

        <title>
        <meta>
        <link>
        <script>
    </head>

    <body>
        ...
    </body>
</html>
```

Listing 4.7: Other Script 1

```
<!DOCTYPE html>
<html>
    <head>
        <!-- Google Analytics -->
        <script></script>
        <!-- End Google Analytics -->









        <title>
        <meta>
        <link>
        <script>
    </head>

    <body>
        ...
    </body>
</html>
```

Listing 4.8: Other Script 2

```
<!DOCTYPE html>
<html>
    <head>
        <!-- Google Analytics -->
        <script></script>
        <!-- End Google Analytics -->

        <!-- Other Script-->
        <script></script>
        <!-- End Other Script -->

        <title>
        <meta>
        <link>
        <script>
    </head>

    <body>
        ...
    </body>
</html>
```

4.2 Test Object: HTML Template / Test website ideas

- Several ideas are proposed
- Each idea has pro and contra: each idea should be discussed of its usefulness, advantages and disadvantages

4.2.1 WordPress

- Show usage of WordPress with some statistics: Why is it so verbreitet
- Explain Plugin system
- Explain Setup on localhost with wocommerce and GA plugin
- Elaborate why this idea was not used

4.2.2 Plain / Skeletal Website

- Idea: Lab environment to have control over all and see effects of changing independent variables
- Problem: Too far away from reality
- Use this as the simplest test possible, not even POC (POC is http archive site)

4.2.3 HTTP Archive inspired website

- Idea: Get correct page weight
- POC: Show that changing independent variables X affect result

4.2.4 Mirroring a complete e-commerce website

- Which website / shop to clone? Show some statistics about biggest e-commerce websites in germany

Otto Re-write this to otto start page clone chapter

Manual adjustments: - Move everything to test folder because top domain is /otto

What did not work (mostly 404s): - user-set-consent-id-cookie: Cookie with name consentId is not set, user-set-consent-id-cookie returns therefore 404 - subscribeToNewsletterSnippetContent: Change path did not work... - amount.json: Not found, also wl_miniWishlistAmount in local storage does not created - a_info: Mock a_info response json does not work...

- footer - userTiming

WPT RV is returning empty csv when 404s are encountered. Therefore i mock the missing ressources so that WPT can run bulk tests successfully.

- mock image sprite_all_1ba408b2.png
- create empty file called user-set-consent-id-cookie
- change path for subscribeToNewsletterSnippetContent: This will remove the cookie banner... but then WPT works

- Idea: Close to reality as possible
- Problems when mirroring a website
- Elaborate why this idea of mirroring complete website was not used
- I used mock of start page of otto, which works fine
- Compare original otto website with mock

Comparison to original webpage

- Remove GA again from mock, so that mock and original are as similar as possible
- Run the same lab test on both pages: WPT and mabye lighthouse
- Compare both results and explain differences

- Setup: Run WPT on mock and on original website - WPT config: - Browser: Chrome - Number of test runs: 3 - FV and RV - Capture Video - Capture DevTools Timeline - Bulk testing: 100x

Diagrams with FV and RV for both cases:
Technical: - First Byte - Bytes In (Doc) - Requests (Doc)
VIsual: - Document Complete - Speed Index

Problem with Repeat View - Problem with RV, Caching: Otto sets request headers to cache-control: no-cache which means that RV basically downloads all resources again. The mock is hosted on Github, where the cache-control header is set to ... It is not possible to change the github request headers. We can modify the http request headers via html, but this is not a clean solution. Therefore I use a different e-commerce website which does not shut down caching so that the RV results are more similar.

Ideally I would host the mock website on a similar infrastructure as the original site with the same webserver configuration. This is for a masters thesis not feasible.

Zalando Idea: It looks like zalando page does not has that many cache-control headers, therefore it may be easier to clone so that RVs are more similar.

Comparison Diagrams with fixed traffic shaping:

4.3 Test Runs

- This section covers all conducted test runs
- Explain test configuration: how many runs, dependent and independent variables, etc.

4.3.1 WPT Configurations

General Settings

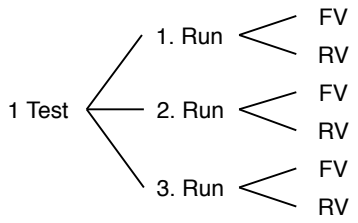
Table 4.1: Test Runs [Sch99]

Configuration Setting	Options	GA
Test Location	Test Location	.
Browser	Firefox, Chrome	.
Connection	LAN	.
Number of Tests to Run	1 to 9	..
Repeat View	First View and Repeat View, First View Only	.
Capture Video	True or False	..
Keep Test Private	True or False	...
Label	Any String	...
Advanced Tab	...	...
Chromium Tab	...	...
Auth, Script, Block, SPOF, Custom Tabs	...	...
Bulk Testing Tab	List of URLs	...

Explanations First View: "First View refers to the cold cache setup in which nothing is served locally" Repeat View: "Repeat View refers to the warm cache containing everything instantiated by the first view" (2016 Using WPT p. 62)

Capture Video: ...

Figure 4.1: Number of tests to run: 3, First View and Repeat View



For one test, we have actually six times that the website gets loaded and tested. For e.g. 500 URLs in the bulk test list, we have a total of $500 \times 6 = 3000$ page hits.

Configuration 1

Table 4.2: Configuration 1

Configuration Setting	Option
Test Location	Test Location
Browser	Chrome
Connection	LAN
Desktop Browser Dimensions	default (1366x768)
Number of Tests to Run	1
Repeat View	First View and Repeat View
Capture Video	True
Keep Test Private	False
Label	none
Advanced Tab	Nothing selected
Chromium Tab	Capture Dev Tools Timeline selected
Auth, Script, Block, SPOF, Custom Tabs	Nothing
Bulk Testing Tab	Test URL x times according to test plan

Configuration 2 Emulate Mobile Browser

Traffic Shaping

- Important to have stable and realistic network condition
- Chromes tool is not the best for this
- Private WPT Instance docker on mac does not allow traffic shaping functionality from WPT
- I use Network Link Conditioner from Apple to slow down the whole machine. See in same blogpost that Patrick highly recommends this
- WPT also slows down their whole machines
- IN general internet connection is very unstable. If i run network link conditionier with e.g. DSL each speedtest gives different results. And other test platforms such as fast.com gives also different result.
- as long as internet connection is stable along all tests, it should not make a big difference because i compare the different variants. Therefore internet connection will fall out of the equation
- i will use the durchschnitt in germany which seems to be 40 mbit per second. or actually i use LTE profile from network conditioner which is 50 mbit per second

4.3.2 Test Object (Website) Variations

as described before, i will compare the values within one independent variable. This is needed in order to compare the impact of the different values within one IV. For example,

i want to measure if there is a difference in performance between the different script attributes. To measure this, i set the default values for the other IVs and vary the values for the IV attribute

Positions: 1: Top of head element 2: just before closing head element 3: just before closing body element

Variants

Variant	Position	Attribute	Other Scripts
Variant P1	top-head	none	no
Variant P2	bottom-head	none	no
Variant P3	bottom-body	none	no
Variant A1	top-head	none	no
Variant A2	top-head	async	no
Variant A3	top-head	defer	no
Variant OS1	top-head	none	no
Variant OS2	top-head	none	yes

Table 4.3: Your caption here

I will not compare variants which are not from the same subgroup, e.g. Variant A2 will not be compared to Variant OS2. Because the first row of the variants table also includes the default values for Attribute and Other Scripts, VP1 is equal to VA1 and VO1.

With the defined IVs and variants, I can create the test objects, that is the index.html files with the corresponding setup. Because its easier to differentiate i will create for the three equal variants nevertheless own index files.

For each test variant, I will create a concrete test artefact, which is a modified index.html. This index.html needs to be uploaded to the webserver before starting with the tests.

All variants will have the same name which is index.html. This is the default file which will be delivered by the webserver when accessing root path of webpage.

Variants to measure: ————————

- Original website - Mock without GA - Position 1 - Position 2 - Position 3 - Attribute 1 - Attribute 2 - Attribute 3 - Other Script True - Other Script False

4.3.3 Test Plan. Generate the data

The Google Analytics code is more or less fixed and there are no configurations. It would be possible to change config of script, e.g. change sample rate, track other metrics etc. But it is not possible to change default tracking behaviour (?)

How the script is included into the file should reflected withing Website Variations

I will use only one WPT Configuration for all tests. Other WPT config can be used in future work, e.g. emulate mobile device.

Table 4.4: Test Runs [Sch99]

Variant	Traffic Shaping	Runs	Date
V-P1	DSL	500	2021-05-07
V-P2	DSL	500	2021-05-07
V-P3	DSL	500	2021-05-07
V-A1	DSL	500	2021-05-07
V-A2	DSL	500	2021-05-07
V-A3	DSL	500	2021-05-07
V-OS1	DSL	500	2021-05-07
V-OS1	DSL	500	2021-05-07

Pre-step: Compare Mock website with and without GA included The comparison between mock and original is part of chapter Test Object

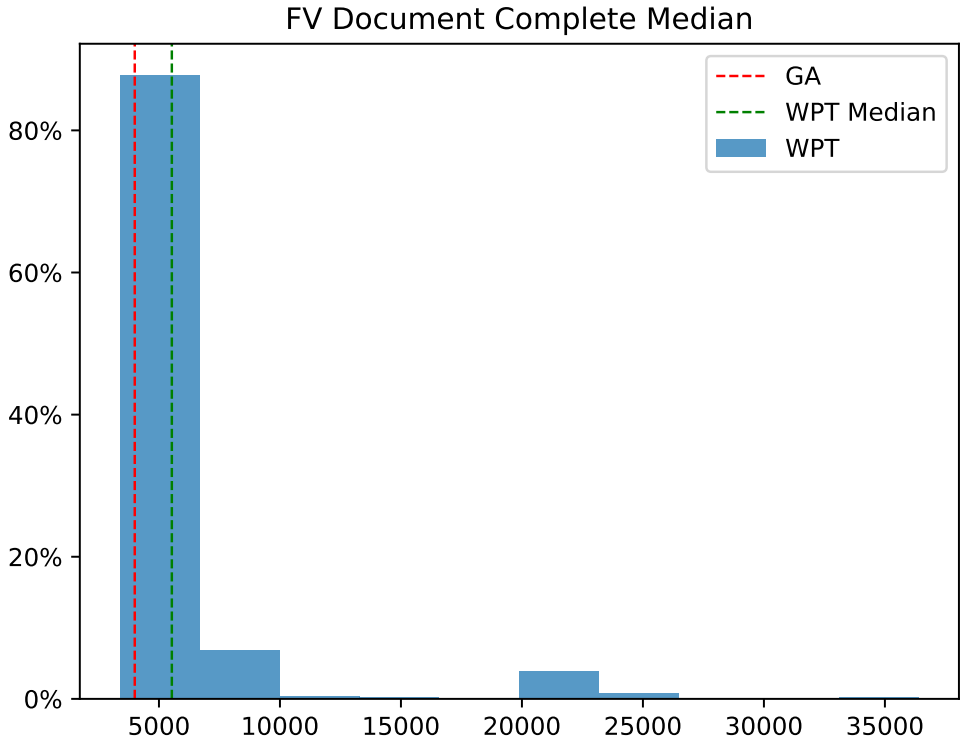
4.3.4 Test Protocol

- Deploy variant of index.html by pushing to GitHub
- Start Network Link Conditioner with specified config on local machine
- Test internet speed with speedtest-cli
- Start local WPT server and agent
- Configure WPT according to specified setup and add list of urls to bulk test interface
- Run test
- When finished, download summary csv file
- On GA helper site, fetch and download data for the current day

4.3.5 Tool support for diagrams and data analysis

- python
- Matplotlib
- seaborn library

5 Evaluation



- Last chapter...
- This chapter: Describe shortly all sections from this chapter
- In the next chapter...

5.1 Test Results

5.1.1 Metrics for Evaluation

Page Weight: Measured by WPT: - bytes - bytes uncompressed - Requests

Technical Timings / API: Measured by WPT and GA: - page load time - domain lookup time - page download time - redirection time - server connection time - server response time - Dom interactive time - Dom content loaded time

Visual Metrics / Web Vitals: Measured by WPT TODO Measure also with GA / own script ??: - CLS - FCP - FMP - LCP - SI

- Visually complete ? - Time to Interactive ? Is this the same as DOM interactive time ? Core Web Vital FID ?? -> Can not be measured without real users... From WPT bulk section. Also include this somewhere for comparison ?: - Filmstrips ? - Waterfall ? - Visual Progress ? - Layout Shifts ?

5.1.2 Original vs Mock Plain

5.1.3 Mock Plain vs Position 1 (which is default position of GA: Check this again!)

TODO rename this like with GA true false ?

5.1.4 Position 1 vs 2 vs 3

5.1.5 Attribute 1 vs 2 vs 3

5.1.6 Other script True vs False

5.2 General

- For each attempt, describe: Threats to validity, generalizability

generalizability: meine Daten zeige nur für Chrome, MacBook, diese Geschwindigkeit etc. Und auch nur für diese Test-Website Die Schwierigkeit der Generalisierbarkeit ist eines der grössten Probleme bei dieser Fragestellung

5.3 Plain / Skeletal Website

- Information gained from this experiment
- Limitations and questions which can not be answered with this approach

5.4 Mirroring

5.5 HTTP Archive inspired website

- Information gained from this experiment
- Meaning and interpretation of the collected data
- Limitations and questions which can not be answered with this approach

5.6 WebPageTest Bulk Tests

- Bulk testing is a feature for private instances only
- Misuse this feature to test the same website X times

5.6.1 Bulk Test Overview: Description of test result page

- Each test has Test ID: YYMMDD_random_random
- Test results after bulk test available under `http://localhost:4000/result/{testID}/`
- For each test run, following data is available:
  - Link to test results: Test result page as same as for single test run
  - Median load time (First view)
  - Median load time (Repeat view)
  - Median Speed Index (First View)
  - Raw page data (file: [TestID_summary.csv]
  - Raw object data (file: [TestID_details.csv])
  - Http archive (.har) (file: json)
- Average First View Load Time
- Average Repeat View Load Time
- Combined Raw: Page Data (file: [TestID_summary.csv])
- Combined Raw: Object Data (file: [TestID_details.csv]). For 100 test runs, this file is appr. 20 MB, 24432 rows, 76 columns.
- Aggregate Statistics (file: [TestID_aggregate.csv])

5.6.2 Summary File for one Test

- Contains 6 rows: 3 test runs: for each test runs 1x first view and 1x repeat view
- Rows 1, 3, 5 contain FV, rows 2, 4, 6 contain data for RV

5.6.3 Aggregate Statistics File

- Contains aggregated data from bulk test
- One row for each test run: For 100 URLs in bulk test will be 100 rows in csv
- Each metric is available with Median, Average, Standard Deviation, Min, Max

- Metrics are available once from FV and once for Repeat View
- Metrics:
  - Successful Tests
  - Document Complete
  - Fully Loaded
  - First Byte
  - Start Render
  - Bytes In (Doc)
  - Requests (Doc)
  - Load Event Start
  - Speed Index
  - Last Visual Change
  - Visually Complete
- => For metric details, see Terms and Definitions

5.6.4 Compare Section

WPT has a feature to compare multiple tests. Accessible under compare URL: `http://localhost:4000/video/compare.php?tests={TestID},{TestID},...`

The compare page contains:

- Film strip
- Waterfall diagram
- Visual Progress diagram
- Timings diagram:
  - Visually Complete (First View Visually Complete Median)
  - Last Visual Change
  - Load Time (onload)
  - ...
- Cumulative Layout Shift diagram
- Requests diagram
- Bytes diagram
- Visually complete

- Last Visual Change
- Load Time (onload)
- Load Time (Fully Loaded)
- DOM Content Loaded
- Speed Index
- Time to First Byte
- Time to Title
- Time to Start Render
- CPU Busy Time
- 85% Visually Complete
- 90% Visually Complete
- 95% Visually Complete
- 99% Visually Complete
- First Contentful Paint
- First Meaningful Paint
- Largest Contenful Paint
- Cumulative Layout Shift
- html Requests
- html Bytes
- js Requests
- js Bytes
- css Requests
- css Bytes
- image Requests
- image Bytes
- flash Requests
- flash Bytes

- font Requests
- font Bytes
- video Requests
- video Bytes
- other Requests
- other Bytes

5.7 Internal, external validity

- At this point, i have the data collected and can analyse it
- The quality and quantity of the data needs to be discussed
- Quality: There are chances that some data are malformed, e.g. because internet connection was bad, etc.
- Quantity: Is the amount of data sufficient to make the evaluation generalisable

6 Future Work

- Last chapter...
- This chapter: Describe shortly all sections from this chapter
- In the next chapter...

6.1 Limitations of this thesis

- Discussion of unobserved topics
- Discussion of possible next steps

6.2 Other measurement tools and metrics

- List of tools and metrics worth investigating

6.2.1 Google Analytics 4

6.3 Speed Kit

6.4 PWAs, AMPs, Service Workers, Caching, HTTP2 etc.

- Overview of other web technologies and how they could be relevant for further research

7 Conclusion

- Last chapter...
- This chapter: Describe shortly all sections from this chapter
- Scope and contribution of this thesis
- Short summary of each chapter:
  - Problem statement and why it is worth to examine research question
  - Terms and definitions
  - (Related work)
  - Approach and evaluation of practical work
  - Future work

- Several topics wurden bearbeitet in this thesis, such like mocking a website for testing purposes, literature review, metrics taxonomy, and the main part which is an experiment

8 Appendix

8.1 WebPageTest Bulk Tests

8.1.1 Single Test Raw page data

WPT Metrics from summary file

Name	Description
minify_total	Total bytes of minifiable text static assets.
responses_200	The number of responses with HTTP status code of 200, OK.
testStartOffset	...
bytesOut	The total bytes sent from the browser to other servers.
gzip_savings	Total bytes of compressed responses.
requestsFull	...
start_epoch	...
connections	The number of connections used.
base_page_cdn	The CDN provider for the base page.
bytesOutDoc	Same as bytesOut but only includes bytes until the Document Complete event. Usually when all the page content has loaded (window.onload).
result	Test result code.
final_base_page_request_id	...
basePageSSLTime	...
docTime	Same as loadTime.
domContentLoadedEventEnd	Time in ms since navigation started until document DOMContentLoaded event finished.
image_savings	Total bytes of compressed images.
requestsDoc	The number of requests until Document Complete event.
firstMeaningfulPaint	...
score_cookies	WebPageTest performance review score for not using cookies on static assets.
firstPaint	RUM First Paint Time, the time in ms when browser first painted something on screen. It's calculated on the client for browsers that implement this method.
score_cdn	WebPageTest performance review score for using CDN for all static assets.
optimization_checked	Whether or not optmizations were checked.

score_minify	WebPageTest performance review score for minifying text static assets.
gzip_total	Total bytes of compressible responses.
responses_404	The number of responses with HTTP status code of 404, not found.
loadTime	The total time taken to load the page (window.onload) in ms.
URL	The tested page URL.
score_combine	WebPageTest performance review score for bundling JavaScript and/or CSS assets.
firstContentfulPaint	...
image_total	Total bytes of images.
score_etags	WebPageTest performance review score for disabling *ETag*s.
loadEventStart	Time in ms since navigation started until window.onload event was triggered (from W3C Navigation Timing).
minify_savings	Total bytes of minified text static assets.
score_progressive_jpeg	WebPageTest performance review score for using progressive JPEG.
domInteractive	...
score_gzip	WebPageTest performance review score for using gzip compression for transferring compressable responses.
score_compress	WebPageTest performance review score for compressing images.
domContentLoadedEventStart	Time in ms since navigation started until document DOMContentLoaded event was triggered (from W3C Navigation Timing).
final_url	...
bytesInDoc	Same as bytestIn but only includes bytes until Document Complete event.
firstImagePaint	...
score_keep-alive	WebPageTest performance review score for using persistent connections.
loadEventEnd	Time in ms since navigation started until window.onload event finished.
cached	0 for first view or 1 for repeat view.
score_cache	WebPageTest performance review score for leveraging browser caching of static assets.
responses_other	The number of responses with HTTPS status code different from 200 or 404.
main_frame	...
fullyLoaded	The time (in ms) the page took to be fully loaded — e.g., 2 seconds of no network activity after Document Complete. This will usually include any activity that is triggered by javascript after the main page loads.
requests	List of details of all requests on tested page.

final_base_page_request	...
TTFB	Time to first byte, which is the duration in ms from when the user first made the HTTP request to the very first byte of the page being received by the browser.
bytesIn	The amount of data that browser had to download in order to load the page. It is also commonly referred to as the page size.
osPlatform	...
test_run_time_ms	...
tester	The ID of tester that performed the page test.
browser_version	The browser version.
document_origin	...
document_URL	...
date	Time and date (number of seconds since Epoch) when test was complete.
PerformancePaintTiming.first-paint	...
osVersion	...
domElements	The total number of DOM elements.
browserVersion	The browser version.
fullyLoadedCPUms	CPU busy time in ms until page was fully loaded.
browser_name	The browser name.
PerformancePaintTiming.first-contentful-paint	...
base_page_cname	...
eventName	...
os_version	...
base_page_dns_server	...
fullyLoadedCPUpct	Average CPU utilization up until page is fully loaded.
domComplete	...
base_page_ip_ptr	...
document_hostname	...
lastVisualChange	Time in ms until the last visual changed occured.
visualComplete	Time in ms when page was visually completed.
render	The first point in time (in ms) that something was displayed to the screen. Before that user was staring at a blank page. This does not necessarily mean the user saw the page content — it could just be something as simple as a background color — but it is the first indication of something happening for the user.
SpeedIndex	The SpeedIndex score.
visualComplete85	Time in ms when page was visually completed 85%.
visualComplete90	Time in ms when page was visually completed 90%.
visualComplete95	Time in ms when page was visually completed 95%.
visualComplete99	Time in ms when page was visually completed 99%.
LargestContentfulPaintType	...
LargestContentfulPaintNodeType	...

chromeUserTiming.navigationStart	...
chromeUserTiming.fetchStart	...
chromeUserTiming.responseEnd	...
chromeUserTiming.domLoading	...
chromeUserTiming.markAsMainFrame	...
chromeUserTiming.domInteractive	...
chromeUserTiming.domContentLoadedEventStart	...
chromeUserTiming.domContentLoadedEventEnd	...
chromeUserTiming.firstPaint	...
chromeUserTiming.firstContentfulPaint	...
chromeUserTiming.firstImagePaint	...
chromeUserTiming.firstMeaningfulPaint	...
chromeUserTiming.firstMeaningfulPaintCandidate	...
chromeUserTiming.domComplete	...
chromeUserTiming.loadEventStart	...
chromeUserTiming.loadEventEnd	...
chromeUserTiming.LargestContentfulPaint	...
chromeUserTiming.LargestTextPaint	...
chromeUserTiming.CumulativeLayoutShift	...
run	The run number.
step	...
effectiveBps	Bytes per seconds, i.e.: total of bytes in / total time to load the page.
effectiveBpsDoc	Same as effectiveBps but until Document Complete event.
domTime	The total time in ms until a given DOM element (specified via domelement parameter when running a test) was found on the page.
aft	Above the Fold Time (no longer supported). The time taken to load everything in the viewport above the fold.
titleTime	Total time in ms until page title was set on browser.
domLoading	...
server_rtt	...
smallImageCount	...
bigImageCount	...
maybeCaptcha	...
bytes.html	...
requests.html	...
bytesUncompressed.html	...
bytes.js	...
requests.js	...
bytesUncompressed.js	...
bytes.css	...
requests.css	...
bytesUncompressed.css	...
bytes.image	...
requests.image	...

bytesUncompressed.image	...
bytes.flash	...
requests.flash	...
bytesUncompressed.flash	...
bytes.font	...
requests.font	...
bytesUncompressed.font	...
bytes.video	...
requests.video	...
bytesUncompressed.video	...
bytes.other	...
requests.other	...
bytesUncompressed.other	...
id	...
chromeUserTiming.InteractiveTime	...

Table 8.1: Your caption here

8.1.2 Single Test Raw object data

8.1.3 Single Test Http archive (.har)

8.1.4 Combined Test Raw page data

8.1.5 Combined Test Raw object data

8.1.6 Combined Test Aggregate data

Bibliography

[Abe] ABERDEENGROUP: *The Performance of Web Applications. Customers Are Won or Lost in One Second.* https://info.headspin.io/hubfs/Analyst%20Reports/5136-RR-performance-web-application.pdf, Abruf: 2021-06-06

[Akaa] AKAMAI: *Akamai Online Retail Performance Report: Milliseconds Are Critical.* https://www.akamai.com/de/de/about/news/press/2017-press/akamai-releases-spring-2017-state-of-online-retail-performance-repor jsp, Abruf: 2021-06-06

[Akab] AKAMAI: *Performance Matters. 9 Key Consumer Insights.* https://www.akamai.com/us/en/multimedia/documents/content/akamai-performance-matters-key-consumer-insights-ebook.pdf, Abruf: 2021-06-06

[BAB+20] BHATTI, Anam ; AKRAM, Hamza ; BASIT, Hafiz M. ; KHAN, Ahmed U. ; NAQVI, Syeda Mahwish R. ; BILAL, Muhammad: E-commerce trends during COVID-19 Pandemic. In: *International Journal of Future Generation Communication and Networking* 13 (2020), Nr. 2, S. 1449–1452. – ISSN 2233–7857 IJFGCN

[BGP15] BEKAVAC, Ivan ; GARBIN PRANIČEVIĆ, Daniela: Web analytics tools and web metrics tools: An overview and comparative analysis. In: *Croatian Operational Research Review* 6 (2015), Oktober, Nr. 2, S. 373–386

[CA11] COHEN-ALMAGOR, Raphael: Internet History. In: *International Journal of Technoethics* 2 (2011), April, Nr. 2, 45–64. http://dx.doi.org/10.4018/jte.2011040104. – DOI 10.4018/jte.2011040104. – ISSN 1947–3451, 1947–346X

[CKR] CROCKER, Cliff ; KULICK, Aaron ; RAM, Balaji: *Real-User Monitoring at Walmart. SF & SV Web Performance Group.* https://www.slideshare.net/devonauerswald/walmart-pagespeedslide, Abruf: 2021-06-06

[CP09] CROLL, Alistair ; POWER, Sean: *Complete Web Monitoring*. 1st ed. Beijing; Cambridge [Mass.] : O'Reilly, 2009

[Dev] DEVICEATLAS: *Android v iOS market share 2019*. https://deviceatlas.com/blog/android-v-ios-market-share, Abruf: 2021-06-06

[For] FORRESTER: *The Total Economic Impact Of Accelerated Mobile Pages*. https://amp.dev/static/files/The_Total_Economic_Impact_Of_AMP.pdf, Abruf: 2021-06-06

[GA] GOOGLE ; AWWWARDS: *Speed Matters. Designing for Mobile Performance*. https://www.awwwards.com/brain-food-perceived-performance/, Abruf: 2021-05-06

[Ges21] GESSERT, Felix: *Mobile Site Speed and the Impact on E-Commerce*. https://youtu.be/RTt1RfMUvOQ. Version: 05.06.2021. – code.talks 2019

[Gri13] GRIGORIK, Ilya: *High Performance Browser Networking*. Beijing ; Sebastopol, CA : O'Reilly, 2013

[Hei20] HEINEMANN, Gerrit: *Der neue Online-Handel: Geschäftsmodelle, Geschäftssysteme und Benchmarks im E-Commerce*. Wiesbaden : Springer Fachmedien Wiesbaden, 2020

[Her19] HERMOGENO, Darwin L.: E-Commerce: History and Impact on the Business and Consumers. In: *International Journal of Engineering Science* 9 (2019), Nr. 3

[IKOK10] ISLAM, A.K.M. N. ; KOIVULAHTI-OJALA, Mervi ; KÄKÖLÄ, Timo: A lightweight, industrially-validated instrument to measure user satisfaction and service quality experienced by the users of a UML modeling tool. In: *AMCIS 2010 Proceedings* 8 (2010)

[Jan09] JANSEN, Bernard J.: Understanding User-Web Interactions via Web Analytics. In: *Synthesis Lectures on Information Concepts, Retrieval, and Services* 1 (2009), Januar, Nr. 1, S. 1–102

[KL17] KOHAVI, Ron ; LONGBOTHAM, Roger: Online Controlled Experiments and A/B Testing. In: *Encyclopedia of Machine Learning and Data Mining*. Boston, MA : Springer US, 2017, S. 922–929

[KO20] KUMAR, Vikas ; OGUNMOLA, Gabriel A.: Web Analytics for Knowledge Creation: A Systematic Review of Tools, Techniques, and Practices. In: *International Journal of Cyber Behavior, Psychology and Learning* 10 (2020), Januar, Nr. 1, S. 1–14

[Lin] LINDEN, Greg: *Make Data Useful. Standford Data Mining Class CS345A, 2006.* http://glinden.blogspot.com/2006/12/slides-from-my-talk-at-stanford.html, Abruf: 2021-06-06

[LO20] LANG, Gil ; OTTEN, Steffen: *Erfolgreicher Online-Handel für Dummies*. Weinheim : Wiley-VCH, 2020

[MAC] MEDER, Sam ; ANTONOV, Vadim ; CHANG, Jeff: *Driving user growth with performance improvements*. `https://medium.com/pinterest-engineering/driving-user-growth-with-performance-improvements-cfc50dafadd7`, Abruf: 2021-06-06

[Mar15] MAREK, Kate: *Library technology reports: expert guides to library systems and services. Using Web Analytics in the Library. Volume 47, Number 5*. 2015

[May] MAYER, Marissa: *Conference Keynote, Web 2.0, 2006*

[Mor18] MORYS, André: Mit A/B-Tests die Optimierungsideen validieren. In: *Die digitale Wachstumsstrategie*. Wiesbaden : Springer Fachmedien Wiesbaden, 2018, S. 97–119

[Mos] MOSES, Lucia: *How GQ cut its webpage load time by 80 percent*. `https://digiday.com/media/gq-com-cut-page-load-time-80-percent/`, Abruf: 2021-06-06

[NC11] NAKATANI, Kazuo ; CHUANG, Ta-Tao: A web analytics tool selection method: an analytical hierarchy process approach. In: *Internet Research* 21 (2011), Januar, Nr. 2, S. 171–186

[SB19] STONE, Brad ; BEZOS, Jeffrey: *Der Allesverkäufer: Jeff Bezos und das Imperium von Amazon*. 2. erweiterte Neuausgabe. Frankfurt : Campus Verlag, 2019

[SBR+19] STEIREIF, Alexander ; BÜCKLE, Markus ; RIEKER, Rouven ; ERTLER, Bernhard ; SKIBBA, Janina: *Handbuch Online-Shop: Strategien, Erfolgsrezepte, Lösungen*. 2., aktualisierte und erweiterte Auflage. Bonn : Rheinwerk Verlag, 2019 (Rheinwerk Computing)

[Sch99] SCHWARZ, Kerstin: *DISDBIS*. Bd. 64: *Das Konzept der Transaktionshülle zur konsistenten Spezifikation von Abhänigigkeiten in komplexen Anwendungen*. Infix Verlag, St. Augustin, Germany, 1999

[SKG+] SCHELHOWE, Luetke ; KAGAWA, Shuhei ; GRUDA, Thorbjoern ; CYBULSKI, Jeff ; MARTIN, David: *Loading Time Matters*. `https://engineering.zalando.com/posts/2018/06/loading-time-matters.html`, Abruf: 2021-06-06

[SKS14] SINGAL, Himani ; KOHLI, Shruti ; SHARMA, Amit K.: Web analytics: State-of-art & literature assessment. In: *2014 5th International Conference - Confluence The Next Generation Information Technology Summit (Confluence)*. Noida : IEEE, September 2014, S. 24–29

[SSMG17] SANTOS, Valdeci Ferreira d. ; SABINO, Leandro R. ; MORAIS, Greiciele M. ; GONCALVES, Carlos A.: E-Commerce: A Short History Follow-up on Possible Trends. In: *International Journal of Business Administration* 8 (2017), Oktober, Nr. 7, 130. http://dx.doi.org/10.5430/ijba.v8n7p130. – DOI 10.5430/ijba.v8n7p130. – ISSN 1923–4015, 1923–4007

[Unb] UNBOUNCE: *Think Fast: The 2019 Page Speed Report Stats & Trends for Marketers*. https://unbounce.com/page-speed-report/, Abruf: 2021-05-06

[Wik16] WIKIPEDIA: *Wissenschaftliche Arbeit*. https://de.wikipedia.org/w/index.php?title=Wissenschaftliche_Arbeit&oldid=156007167. Version: 2016, Abruf: 2016-07-21

[WK09] WAISBERG, Daniel ; KAUSHIK, Avinash: Web Analytics 2.0: Empowering Customer Centricity. (2009)

[ZP15] ZHENG, Guangzhi ; PELTSVERGER, Svetlana: Web Analytics Overview. In: KHOSROW-POUR, Mehdi (Hrsg.): *Encyclopedia of information science and technology*. Information Science Reference, 2015, Kapitel 756, S. 7674–7683

Eidesstattliche Versicherung

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe angefertigt und mich anderer als der im beigefügten Verzeichnis angegebenen Hilfsmittel nicht bedient habe. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen wurden, sind als solche kenntlich gemacht. Ich versichere weiterhin, dass ich die Arbeit vorher nicht in einem anderen Prüfungsverfahren eingereicht habe und die eingereichte schriftliche Fassung der auf dem elektronischen Speichermedium entspricht.
Ich bin mit einer Einstellung in den Bestand der Bibliothek des Fachbereiches einverstanden.

Hamburg, den _____ Unterschrift: _____