UNIVERSITÀ
DI PAVIA

EDI: Third/Fourth Lab Report

D. Ligari 518592[1]

[1] *University of Pavia, Department of Computer Engineering (Data Science), Pavia, Italy*
Contact: davide.ligari01@universitadipavia.it
Date: May 29, 2023

Abstract— This report examines the impact of web technologies on Page Load Times (PLTs) for commercial and institutional websites. It analyzes parallel connections, caching policies, and performance evaluation tools. The findings highlight the significance of parallel connections, the effects of caching policies on PLTs, and provide insights for optimizing website performance. The report also evaluates website performance under different conditions and explores the role of warm-up time.

Keywords—HTTP cache policies • HTTP • PLT • Apache HTTP server benchmarking tool • h2load

CONTENTS

1. WEB TECHNOLOGIES

In today's digital landscape, web technologies play a pivotal role in determining the performance and overall user experience of modern websites. As the internet continues to evolve and user expectations rise, it becomes increasingly crucial to understand the profound impact that these technologies have on Page Load Times (PLTs) for commercial and institutional websites.
This report aims to delve into various aspects of web technologies, exploring their effects on PLTs and providing valuable insights to optimize website performance.

a. Impact of parallel connections on PLTs

Analyze and discuss the impacts of the number of parallel connections set inside the browser on the Page Load Times of commercial/institutional websites. Did you notice any expected or unexpected behavior?

The impact of increasing the number of connections on page loading time is evident and consistent, as depicted in Figure 1. Experimental results clearly demonstrate that as the number of connections increases, the loading time consistently decreases, showcasing the advantage of parallel downloading. This effect can be attributed to the browser's ability to simultaneously download multiple files, resulting in an overall improvement in download speed.
However, it's important to note that the relationship between the number of connections and loading time is influenced by the size of the webpage. For smaller page sizes, exemplified by the website *www.pedranzini.it/*, the loading time remains relatively consistent regardless of the number of connections. This suggests that the page can be efficiently loaded with a limited number of connections due to its compact size. The downloading process for such pages is optimized, and further increasing the number of connections does not significantly impact the loading time.
Conversely, for larger page sizes, such as the case of *www.vallespluga.it/*, the loading time exhibits a significant decrease as the number of connections increases. This indicates that the page benefits from parallel connections to efficiently download its extensive content. As the page size increases, the impact of parallel connections on reducing loading time becomes more pronounced.
To provide additional context, the table below presents the websites used for the analysis, along with their respective characteristics.

Website	Page size(KB)	Cache policy	HTTP version	HTTPS
www.pedranzini.it/	180.40	Validation	1.1	Not supported
www.unica.it/	2570	Validation	1.1	Supported
www.uniss.it/	2540	Validation • Expiration	1.1	Supported
www.vallespluga.it/	20120	Validation	1.1	Supported

Table 1: Website used for the analysis

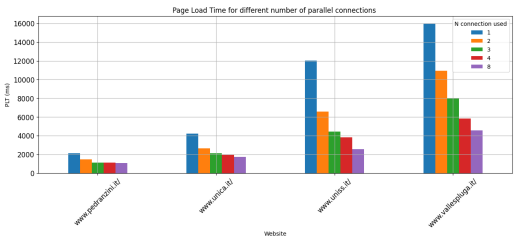

Fig. 1: Page load time vs number of concurrent connections for different sites

b. Impact of caching policies on PLTs

Analyze and discuss the impacts of caching policies implemented by different commercial/institutional websites on the Page Load Times. Consider websites that support HTTP/1.1, HTTP/2 and HTTP/3 (possibly with unsecure and secure connections). Did you notice any expected or unexpected behavior?

To provide a comprehensive understanding, an extensive analysis was conducted on multiple websites with varying characteristics, as presented in Table 2. Thorough testing was ensured by evaluating sites that supported multiple versions of the HTTP protocol with each version.
Upon examining the results depicted in Figure 2, a noteworthy observation emerges: websites supporting HTTP/2 or HTTP/3 consistently exhibit faster loading times compared to those supporting only HTTP/1.1, regardless of the implemented cache policy. This finding underscores the undeniable performance benefits associated with the adoption of newer HTTP protocol versions.
Furthermore, the analysis sheds light on the impact of different cache policies. Websites supporting both validation and expiration cache policies were found to have shorter loading times compared to those supporting only the validation policy. This outcome can be attributed to the combined advantages of validating cached content and leveraging expiration times to minimize server requests. In contrast, websites lacking any caching mechanism displayed higher Page Load Times (PLTs), underscoring the crucial role of effective caching in optimizing web performance.
Considering that page load time is significantly influenced by page size, and the analyzed pages vary in size, the values displayed in Figure 2 have been normalized accordingly. Additionally, it is important to note that the analyzed sites contain objects served by different servers, resulting in requests using different versions of HTTP and cache policies. To address this issue, each site was assigned the most commonly used HTTP cache policy and protocol.

Website	Page size(KB)	Cache policy	HTTP version	HTTPS
www.pedranzini.it/	180.40	Validation	1.1	Not supported
www.unica.it/	2570	Validation	1.1	Supported
www.uniss.it/	2540	Validation • Expiration	1.1	Supported
www.vallespluga.it/	20120	Validation	1.1	Supported
www.istat.it/	11200	Validation	1.1	Supported
www.unina.it/	2210	Validation	1.1	Supported
www.apache.org/	2010	Validation • Expiration	3	Supported
www.studiocamer.com/	3340	Validation • Expiration	2 • 3	Supported
www.off—white.com/	6790	Validation	1.1 • 2 • 3	Supported
www.google.com	2350	Validation • Expiration	1.1 • 2 • 3	Supported

Table 2: Website used for the analysis

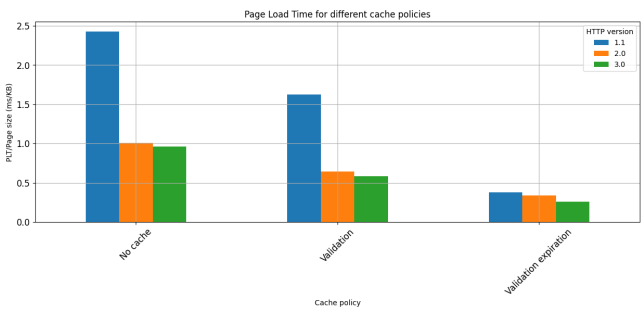

Fig. 2: Page load time over page size for different cache policies for different HTTP versions

c. Performance analisys using Apache HTTP server benchmarking tool

Ab (ApacheBench), it is a command-line tool that comes bundled with the Apache HTTP Server. It is designed to perform load testing and measure the performance of web servers. The "ab" tool allows users to simulate multiple concurrent requests to a web server and record various performance metrics, such as requests per second, latency, and throughput. Below an example of the command used to perform the analysis, where the number of concurrent connections is set to 2 and the number of requests to 30.

```
ab -c 2 -n 30 www.galbusera.it/chi-siamo/
```

Analyze and discuss the performance of different commercial/institutional websites obtained under different conditions using the ab – Apache HTTP server benchmarking tool. Did you notice any expected or unexpected behavior?

The websites listed in Table 3 were subjected to analysis using the 'ab' command to evaluate their performance under different conditions. Specifically, the impact of concurrency level on requests per second and transfer rate was assessed, as depicted in Figure 3 and 4 respectively. The results indicate that increasing the concurrency level leads to higher requests per second and transfer rate, aligning with the findings discussed in Section 1.a, where an increase in the number of connections resulted in decreased page load time. However, there were some unexpected results observed for certain websites. For instance, *www.galbusera.it/chi-siamo/* exhibited a slight decrease in transfer rate and requests per second when transitioning from 4 to 8 connections. This anomaly could be attributed to factors like server congestion or limited network bandwidth at 8 connections. However, with 16 connections, network congestion might be alleviated, potentially improving performance. Alternatively, the website's server infrastructure might incorporate load balancing mechanisms that distribute requests evenly across a limited number of resources, causing a slight decrease in performance with 4 to 8 connections but optimizing resource allocation with 16 connections.
Another anomalous finding was observed for *www.pedranzini.it/*, which displayed a significantly higher number of requests per second compared to other websites but a lower transfer rate. This outcome can be attributed to the smaller size of the website, resulting in a higher number of requests per second despite the server's relatively lower performance in terms of transfer rate.
To gain further insights, an additional analysis was conducted to evaluate the time distribution across the connecting, processing, and waiting phases, as depicted in Figure 5. The results illustrate that the time spent on the connection phase is relatively minimal when compared to the time allocated to processing and waiting.
The graphs clearly demonstrate that the majority of time is dedicated to processing the request on the server and waiting for the necessary operations to be completed. This observation highlights the significance of server-side computations and any external dependencies involved in generating

the response. The processing phase, which encompasses executing server-side scripts, performing database queries, or conducting complex computations, typically consumes a substantial portion of the overall time.
Intriguingly, the waiting time, during which the server may be accessing external resources or performing additional computations before generating the complete response, is found to be lower than the processing time. This suggests that the waiting phase is generally efficient, possibly due to optimized resource access or caching mechanisms employed by the server.

Website	Page size(KB)	Cache policy	HTTP version	HTTPS
www.pedranzini.it/	180.40	Validation	1.1	Not supported
www.vallespluga.it/	20120	Validation	1.1	Supported
www.galbusera.it/chi-siamo/	42147	Validation • Expiration	3	Supported

Table 3: Website used for the analysis

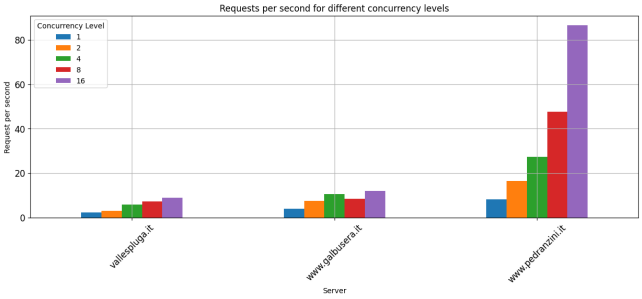

Fig. 3: Number of request per second for different number of concurrent connections for different sites

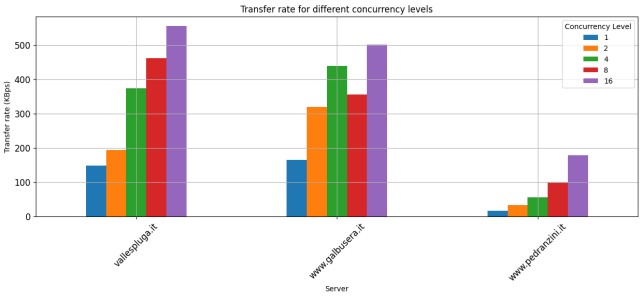

Fig. 4: Transfer rate for different number of concurrent connections for different sites

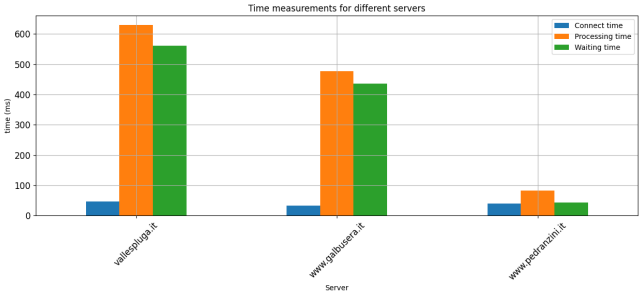

Fig. 5: Time spent for the connection, for processing and for waiting for different sites

d. Performance analisys using h2load

h2load is a command-line benchmarking tool specifically designed for testing the performance of HTTP/2 and HTTP/3 servers. It is part of the nghttp2 project, which focuses on developing and maintaining tools and libraries for working with HTTP/2 and HTTP/3 protocols. h2load allows you to simulate a large number of concurrent requests to a server and measure its performance under various conditions. It can be used to assess the server's capability to handle a high volume of traffic, evaluate response times, measure throughput, and identify potential bottlenecks or performance issues. Below an example of usage.

```
h2load --duration=30 -c 16 --warm-up-time=4  https://
    www.galbusera.it/chi-siamo/
```

e. The role of Warm up time

The warm-up time refers to the initial period during which the load generator gradually ramps up the number of concurrent requests before reaching the desired load level for the test.
The purpose of the warm-up time is to simulate a more realistic user behavior and allow the server to stabilize under increasing load. By gradually increasing the number of concurrent requests, it helps to avoid sudden spikes in traffic that may not accurately represent how users interact with the website.
During the warm-up phase, the load generator starts with a lower number of concurrent requests and gradually increases it over a predefined period of time or until a specific load level is reached. This gradual increase allows the server to adjust to the increasing load, initialize any necessary resources, and optimize its performance before the actual testing phase begins.
The duration of the warm-up time can vary depending on the specific testing scenario and the characteristics of the website being tested. It is typically determined based on factors such as the expected user behavior, the server's response time under increasing load, and any specific performance goals or requirements.
By including a warm-up time in the testing process, the results obtained from h2load or similar tools are more representative of real-world usage patterns, providing a more accurate assessment of the website's performance and the server's ability to handle increasing loads over time.

Analyze and discuss the performance of different commercial/institutional websites obtained under different conditions using the nghttp and h2load tools. In the experiments with h2load analyze the role of the warm-up time. Did you notice any expected or unexpected behavior?

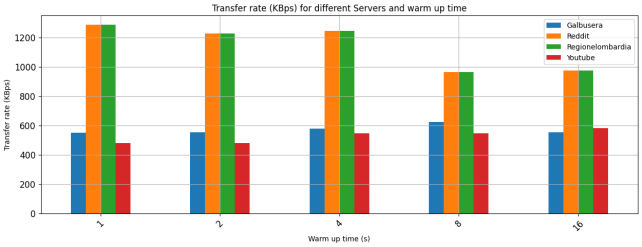

Fig. 6: Transfer rate for different warm up time for different sites

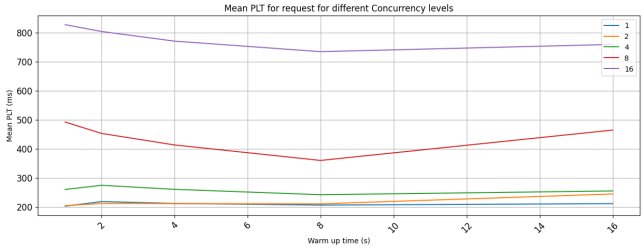

Fig. 7: mean time for different warm up time for different concurrent connections

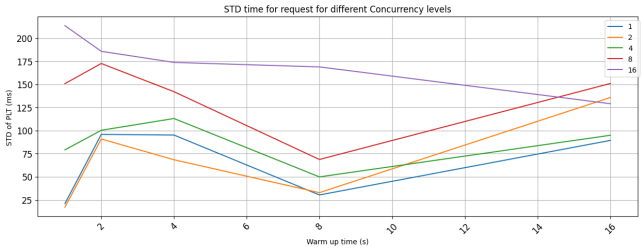

Fig. 8: Standard deviation for different warm up time for different concurrent connections

2. CONCLUSIONS