

# Framework to Model User Request Access Patterns in the World Wide Web

Richard Hurley, Robert Sturgeon

Department of Computer Science, Trent University, Peterborough, ON, Canada

Email: rhurley@trentu.ca, rsturgeon@trentu.ca

**How to cite this paper:** Hurley, R. and Sturgeon, R. (2024) Framework to Model User Request Access Patterns in the World Wide Web. *Journal of Software Engineering and Applications*, 17, 69-88.

<https://doi.org/10.4236/jsea.2024.172004>

**Received:** December 3, 2023

**Accepted:** February 19, 2024

**Published:** February 22, 2024

Copyright © 2024 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## Abstract

In this paper, we present a novel approach to model user request patterns in the World Wide Web. Instead of focusing on the user traffic for web pages, we capture the user interaction at the object level of the web pages. Our framework model consists of three sub-models: one for user file access, one for web pages, and one for storage servers. Web pages are assumed to consist of different types and sizes of objects, which are characterized using several categories: articles, media, and mosaics. The model is implemented with a discrete event simulation and then used to investigate the performance of our system over a variety of parameters in our model. Our performance measure of choice is mean response time and by varying the composition of web pages through our categories, we find that our framework model is able to capture a wide range of conditions that serve as a basis for generating a variety of user request patterns. In addition, we are able to establish a set of parameters that can be used as base cases. One of the goals of this research is for the framework model to be general enough that the parameters can be varied such that it can serve as input for investigating other distributed applications that require the generation of user request access patterns.

## Keywords

Performance Modelling, World Wide Web, Simulation, User Request Access Patterns, Framework

## 1. Introduction

Surfing the Internet has become commonplace in today's society, from finding out the latest sports scores to keeping up on politics to locating recipes. Users are constantly using their devices (cell phones, tablets, laptops, desktops, etc.) to search for information. This search results in a steady stream of user requests for information that can be found on various web pages (see **Figure 1** which presents

a simplified representation of the *World Wide Web* (WWW) that outlines a user making a request to a web server for a web page).

All web pages are not created equal given the wide variety of objects that make up a web page: text, images, audio, video, code, etc. [1]. Given the size and complexity of the WWW, it is challenging to properly capture the behavior of the user request access patterns.

In this paper, we present a framework that models at the object-level, user (*Hypertext Transfer Protocol* (HTTP)) request patterns in the WWW which can then allow one to compare the relative performance of various distributed applications, and the impact these have on user experience. As a framework, our model can be adapted to a wide range of conditions that can be used by researchers as a tool with their own distributed application models such as web-caching [2] [3] [4].

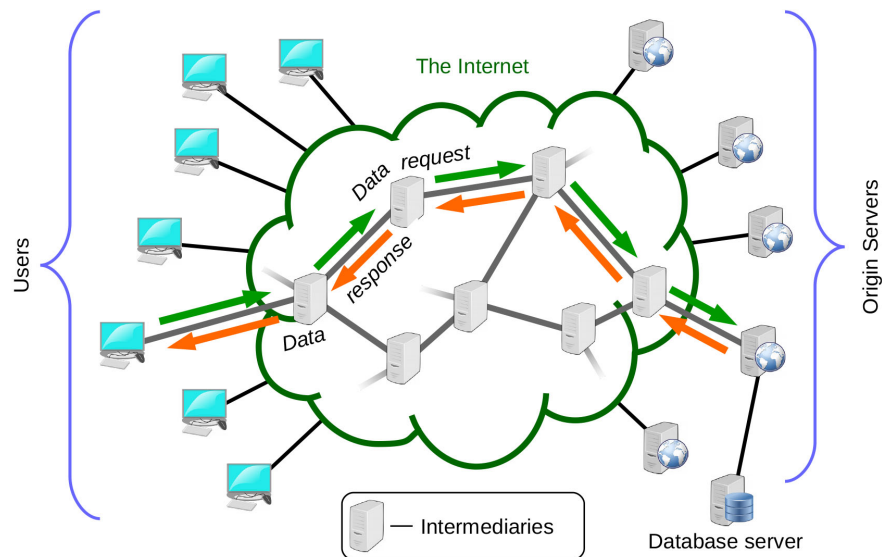
The reality of the WWW is that it is complex: it involves the transmission of large and varied amount of information via the Internet, through a variety of devices spanning many regions, countries, and even into space. Our model attempts to capture much of this and allows for the relative comparison of the performance of various distributed system configurations.

There has not been much past work done on modelling user request access patterns on the web. The main focus in this area has been on analyzing web traffic patterns [5] [6] [7]. Much of the prior work tended to use the web page as the basic unit while we are modelling at the object level [8] [9]. The authors in [10] used 5 years of WWW traffic with over 70,000 users per day corresponding to 1.903 TBytes of traffic. Ihm *et al.* employed their own analysis technique which essentially views the web traffic as a stream of HTTP objects. It is this analysis that we use as a basis for some of our model assumptions.

Our User Request Object-Level Framework Model is actually composed of three sub-models: one which represents the set of users that make requests for web pages as well as how they select web pages, another which consists of a set of web servers that respond to user requests, and finally, one which represents the web pages in the system. This paper is organized as follows: Section 2 gives an overview of the general environment we are examining, Section 3 presents the details of our models, Section 4 is where we show some results of the model, and finally, in Section 5 we present some concluding remarks.

## 2. General Environment

At the highest level, as shown in **Figure 1**, we consider the WWW to consist of the following components: users, web servers, the Internet, and the information being requested. The term *user* will refer an agent that is requesting information—human or machine [11]. The *Internet* is the communications media that interconnects the users and web servers, typically through a series of intermediaries. Finally, *web servers* are the devices that store the information in the form of web pages and transmit results to the user to satisfy a request.



**Figure 1.** Simplified World Wide Web showing an example request/response chain.

Although the main goal of the WWW is to connect users to information, nearly half of the traffic on the WWW is due to automated processes [6]. These machine actors are used for various activities, much of which include information indexing for search engines and analytic algorithms [12]. We do not consider this traffic in our model as we are more concerned with relative performance than absolute performance.

The information available within a WWW varies widely in its amount, size, composition, and format. Some common examples of information types are text, audio, video, and images. Also, these types of information exist in several technological formats, such as plain text or binary information, and may come from a multitude of information sources. Information is accessed by determining its location on the WWW using a *Uniform Resource Identifier* (URI) (for example <https://www.trentu.ca/>). URI are logical addresses for discrete documents, referred to as *web pages* [10].

The composition of a web page can be highly varied, ranging from a monolithic document of plain text, to a complex collection of elements of various forms. Each web page must contain at least one element, but normally consists of many elements called *web page objects*. A web page therefore, can be viewed as a group of web page objects. Web page objects have several properties, but for our model we are primarily concerned with their size as this has a direct effect on the retrieval of the web page. There are dozens of types of web page objects, but some of the common ones that make up a web page are: *Hyper-Text Markup Language* (HTML), Javascript, *Extensible Markup Language* (XML), *Cascading Style Sheets* (CSS), image, audio, video, octet [1] [6].

An important consideration regarding web pages is that their content can change with time (referred to as *dynamic content*) [13]. Although the specific content on a web page may change, it still represents a single resource [14]. For

example, a news main page such as <https://www.cbc.ca/news> consists of a collection of news articles available on the site at some point in time. The collection, however, changes with time as current news articles are added and old ones are removed. Regardless of the news articles available, <https://www.cbc.ca/news> is still viewed as a single web page.

Another component within the WWW is the *web server*, which has an important impact on the experience of the user. Web servers, which range from a single computer to a collection of networked computers, hold the web pages that users request [15]. The speed at which they can respond to the request contributes in large part to the response time. Fundamentally, a web server consists of the following components: an interface to the Internet, processors for requests, storage, and software [16]. There are several ways that these components can be organized within web servers: as a single computer or a collection of multiple machines. For our investigation, we will not need to distinguish between the two.

The WWW would not be as ingrained in society as it is not for the underlying networking technology that now spans most of the world (*i.e.*, the Internet). The Internet is a collection of protocols for communicating across arbitrary packet-switched networks, and is made up of three main components: hosts, routers, and networks [16]. For our study, we will abstract the network delay into the service time for a user request.

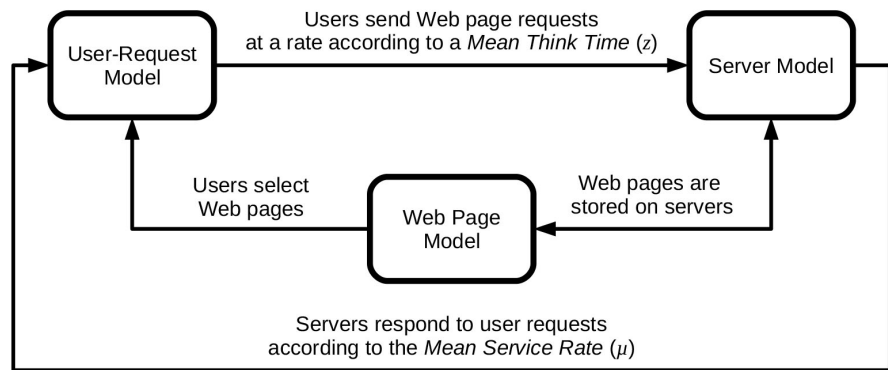
### 3. Model

In this section, we present our framework model that represents the process of users interacting with the WWW to obtain web pages from the respective web servers. A novel feature of our model is that web pages are represented at the object-level. The main performance metric in which we are interested is *mean response time* (MRT): the time from when a user requests a web page until the web page (and all its objects) has been delivered. This performance measure can influence a user's experience substantially [17].

#### 3.1. High-Level Model

The high-level model for user-web interaction is shown in **Figure 2** and represents the total process of a user requesting a web page from a server, and the server's subsequent response. To capture the behavior of the users, servers, and Web pages, the model in **Figure 2** is comprised of three sub-models: *User-Request Model*, *Server Model*, and *Web Page Model*. The User-Request Model manages the state of the users and their requests. Requests are received and processed by web servers within the Server Model. The Web Page Model captures the web pages in terms of their object composition and location.

The interaction of the three sub-models forms the *Request-Response Process*. This process begins with the user selecting a web page using the Web Page Selection Model described in Section 3.2. The web page request is transmitted to



**Figure 2.** User-web iteration model.

the web server that eventually supplies the requested web page to the user. If the web server is currently busy servicing another user request, then the request is placed into a wait queue. Otherwise, the request is serviced immediately. The time to service a request is based on many factors, including the composition of the web page, the number and size of the web page objects, as well as their media type. Once the user receives the response to their request, they cease interacting with the system until their next request (*think time*).

The *think time* controls the rate at which users request web pages. A lower think time leads to a greater load on the web servers. When not in the system waiting for a request to be served, a user is in a think state, which represents a user processing a previously retrieved web page. We assume that the *think time* for users is independent and exponentially distributed, with a mean  $z$ . A user will not request another web page until they have completed their think state.

The performance measure of interest in this research is *response time*: the time from the user initiating the request until receiving the response; with the MRT of all requests that occur in the system being our main performance measure. Response time represents the sum of the transmission latency, the time in queue that the request awaits processing, and the request processing time—which we refer to as *request service time* ( $R_i$ ).

### 3.2. User-Request Model

The User-Request Model represents the process of a user selecting and requesting a web page. This sub-model consists of two main components: the *User Set* and the *Web Page Selection Model*. The purpose of the User Set is to model the users in the system and how they interact with the WWW. We use a finite population model which incorporates the user think time in the request-response process as discussed in Section 3.1 [18].

The Web Page Selection Model represents the mechanism by which users select web pages. The pages stored in a web cache and their request probabilities vary over time. Pages such as a news article, viral videos, course assignments, memes, etc. become popular for periods of time and then eventually become accessed less frequently. To represent this behavior, we use a variant of the dy-

dynamic page reference model described in [19].

The dynamic page reference model for a system with  $M$  web pages is shown in **Figure 3**. This model assumes that a web page can be in one of two states: normal and popular. Web pages in the popular state have a higher request probability than that of the pages in the normal state with  $\nu$  representing the ratio of page requesting rates for the popular to normal state.

The model also assumes that there are two types of pages: conventional and potentially popular. Conventional pages remain in the normal state while potentially popular pages alternate between the normal and popular state based on a continuous-time Markov chain. The rate at which a page transitions from a normal to popular state is  $\lambda_1$  and from popular to normal is  $\lambda_2$  (the time spent in either state is assumed to be exponentially distributed). Finally, we let  $M_0 < M$  denote the number of potentially popular pages that are present in the system. With this type of model, the model is able to generate page requests with high coefficients of variation, a attribute that has historically shown to be desirable in such systems [20].

### 3.3. Server Model

The web servers in our model consist of one or more devices which store the web pages and service the requests. The role of each web server is to process requests and transmit the response back to the user (in the form of the web page and all of its individual objects).

**Figure 4** presents the detail of our Server Model within the User-Web Interaction Model. We assume that web servers can only process one request at a time. If a request is received at a server that is busy, then the incoming request is placed in the server's wait queue (we assume *first-come-first-served* (FCFS), however, this discipline may be modified to suit the desired investigation). Once a server completes a request, it then looks to the wait queue for the next request. If the wait queue is empty, then the server sits idle waiting for the next request.

The *request service time* ( $R_i$ ), the main parameter of the Server Model, is the time to process and retrieve web page  $i$  and transmit it to the user. It follows that the web page size ( $W_i$ ) would have a direct impact on the request service time ( $R_i$ ) as the larger a web page becomes, the longer it takes to service and transmit.

From Equation (1), we can observe that web page size is the sum of the object sizes, where web page  $i$  consists of a unique set of  $Q_i$  objects, and each object has a size  $\Theta_{ij}$ . Thus, the request service time, which is assumed to be exponentially distributed, would have a mean of  $\mu^{-1} W_i$  where  $\mu^{-1}$  is simply the base service time (Equation (2)). The number of objects ( $Q_i$ ) and their object type is an important aspect of our model, and one that can be varied as needed to emulate a plethora of different environments.

$$W_i = \sum_{j=1}^{Q_i} \Theta_{ij}. \quad (1)$$

$$R_i = \mu^{-1} W_i \quad (2)$$

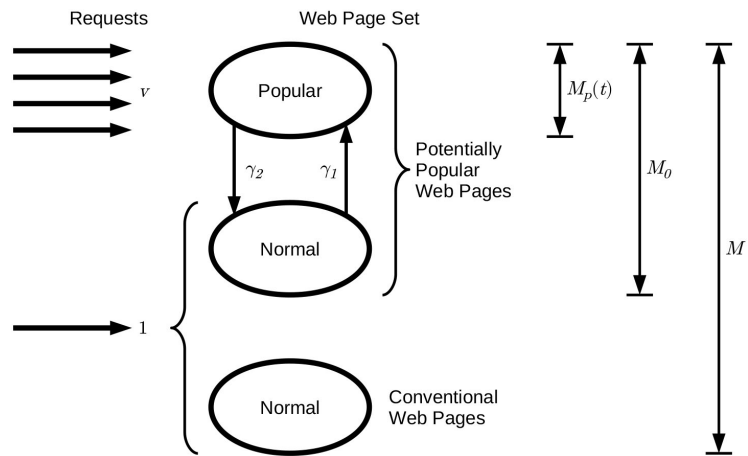


Figure 3. Web page selection model.

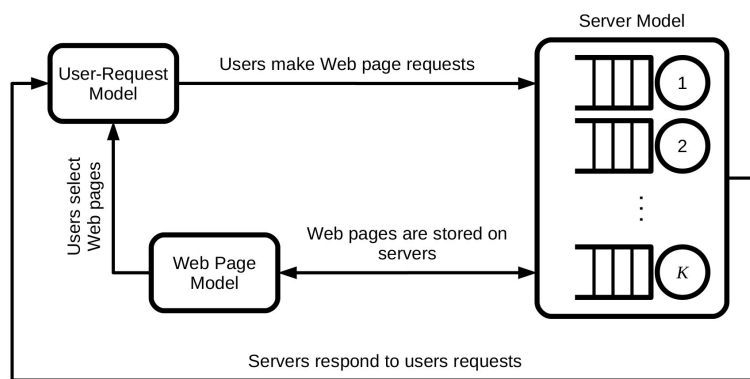


Figure 4. Web page request model, showing FCFS web server queues.

### 3.4. Web Page Model

The web page model represents the totality of information that is requested and delivered to the user. It is modelled as a collection of web page objects that are transmitted from the web server to the user upon request. Each web page object is assumed to be transmitted independent of one another, but a web page request is not fulfilled until all objects that compose the web page are received. Each web page is unique, and consists of the *Web Page Object Set*, which is determined according to a *Web Page Categories*.

#### 3.4.1. Web Page Object Set

A web page consists of a collection of one or more web page objects, which we refer to as the *Web Page Object Set*. The composition of the object set is assumed to be determined by both the number and type of objects. The distribution of the objects and their object types is influenced by the web page category, which is a parameter of the web page and will be discussed shortly.

Each object consists of two main attributes: *object type* and *size*. The *Web Page Object parameters* are based on the work of Ihm ([6] and summarized in **Table 1**. While this set is not exhaustive, these six web page object categories account for the majority of information being transferred via the WWW [6].

**Table 1.** Example web page object parameters.

Object Type	Size	Examples
text	small	HTML, XML, text
script	small, medium	JavaScript
css	small	CSS
image	small, medium	.png, .gif, .jpg
audio	small, medium	.ogg, .mp3, .wmv
video	medium, large	.flv, .mov

From Equation (2), we can observe that object sizes directly affect the web page size, and thus influence the request service time ( $R$ ).

### 3.4.2. Web Page Categories

The next feature of a web page is the *web page category* which characterizes the composition of web pages as defined by our model. The web page category provides a means to define the composition of web page objects within the model by specifying how objects are distributed based on their object type, and size within the object set of each web page. As seen from Equation (2), the category of a web page would directly affect the retrieval time due to the potentially inherent differences in web page sizes.

Using the analysis from [10], we specify three web page categories: *article*, *media*, *mosaic* for our web Page Model but recognize that other page categories and their compositions are certainly possible. The following summarizes our three web page categories:

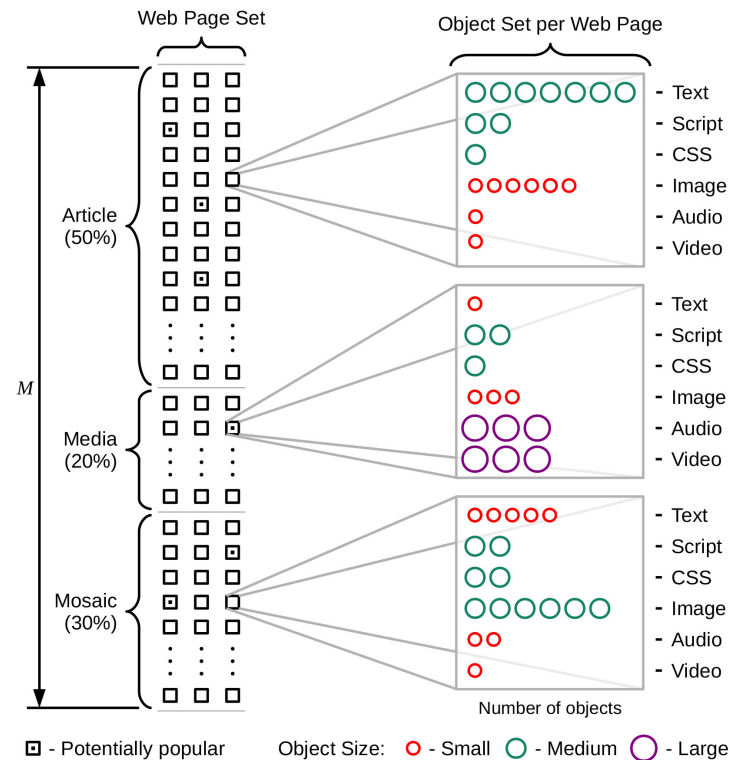
**article** Typically contains information about one specific topic. The composition consists of a large amount of script, text and images, but has very little audio or video information. Examples: news articles, Wikipedia article pages.

**media** While this category is similar to articles, the main difference is that a media page provides more audio and/or video. Examples: image collages such as Google Images, Flickr, Pinterest; image pages, music and video streaming.

**mosaic** Web pages that provide multiple links and summaries to other specific topics. Their size is primarily influenced by text and script, with some image information and a moderate amount of audio and/or video. Examples: search results such as web searches, Pinterest image search; website main pages, social media threads.

The breakdown of web pages into web page categories are input parameters in our model (percentages). In addition, the web pages in each category are configured according to an input table that describes the way in which objects are to be distributed. **Figure 5** shows a graphical example of this configuration for  $M$  web pages and their distribution using the web page categories. It also displays one example web page from each of the three web page categories to illustrate how the page category affects the object distribution.





**Figure 5.** Example web page composition based on category.

From **Figure 5**, we can observe that the web page that is an *article* is dominated by several text objects, each of a medium size, as well as several small images with only a few of each of the other objects. In contrast to an *article* web page, *media* web pages are dominated by a moderate number of large audio and video objects. The *mosaic* web page, however, has several medium sized image objects and a moderate number of other objects.

## 4. Results

Based on the model presented in Section 3, we developed a discrete-event simulation to evaluate the model and investigate the impact of the various parameters. In addition, we establish parameter values that are representative of various loading scenarios that could be used when applying our model to other distributed applications. Details of the implementation, validation and verification of the simulation can be found in [4].

We will begin our investigation in this section by examining the effects of the model parameters for web page composition and determine their relationship to our main parameter of interest, MRT. Several key parameters contribute to the web page composition including web page category, object size, object type, as well as number of objects. With these parameters established, we then examine the impact of system load and coefficient of variation of web page request interarrival times on MRT. We will finish this section with a closer look at the ratio of web page categories in order to determine the effect on performance.

#### 4.1. Composition of the Web Page Set

The *Web Page Set* consists of several parameters that affect the performance of our system, in particular MRT. We refer to the collection of parameters that determine the make-up of the Web Page Set as the *Composition of the Web Page Set*. Ultimately, our goal here is to establish the sizes of the web pages in our system, which vary according to web page categories.

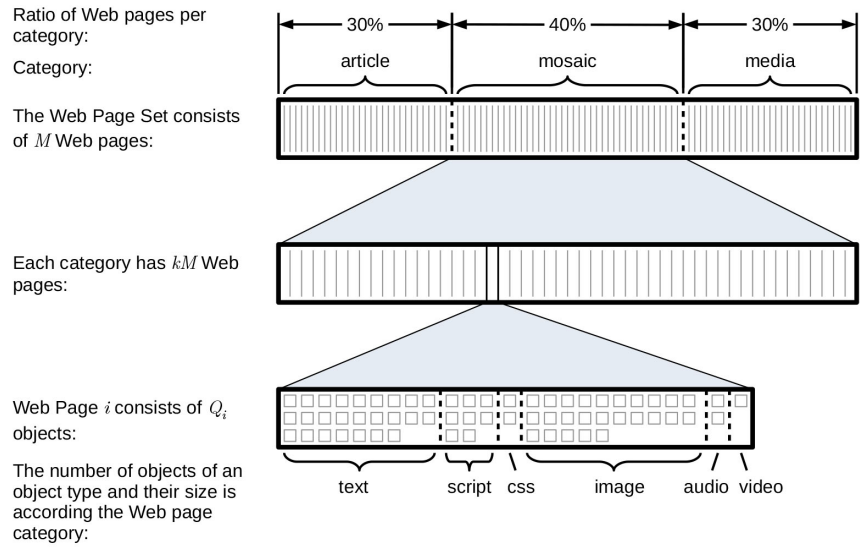
In the *Server* and *Web Page* models introduced in Section 3, we described the parameters that compose the Web Page Set. **Figure 6** provides a visual summary of the relationship between these parameters. The *Web Page Set* consists of  $M$  web pages, where each web page has a composition that follows one of our three web page categories: *article*, *mosaic*, *media*. The number of web pages of each category in our Web Page Set is broken down by the *Ratio of Web Pages per Web Page Category*. Each web page  $i$  has a Web Page Object Set, which consists of  $Q_i$  objects of various types and sizes. The object types and sizes, as well as the quantity for a given web page, are determined by the web page category. The sum of the object sizes in the Object Set determines the size of each web page ( $W_i$ ).

The Ratio of Web Pages per Category describes how many web pages of each category there will be in the Web Page Set. Ratios are expressed as a percentage of  $M$  and are expected to be greater than 0%. We now introduce the term  $k_{\text{category}}$  to be the proportion that a *category* represents in the Ratio of Web Pages per Category. Thus, there are  $k_{\text{category}} M$  Web pages per category.

We next introduce the concept of *Web Page Category Base Scenario* to be the starting point for establishing other model parameters in this section. We assume the following ratio of web pages per category:  $k_{\text{article}} = 30\%$ ,  $k_{\text{mosaic}} = 40\%$ , and  $k_{\text{media}} = 30\%$ . The values for the Base Scenario were chosen to result in the middle of the range of possible mean web page size for all requests [4]. It is important to note that the choices of the parameters for *Web Page Category Base Scenario* could easily be modified as we are only interested in the relative (not absolute) performance of the system. Later in this section, we will investigate the effect that the Ratio of Web Pages per Category has on MRT.

The *Composition of the Web Page Object Set* is important as it establishes the size of each web page. Each category and object type has a categorical object size, and a number of objects. We begin by establishing the categorical Object Sizes ( $\Theta$ ) for each object type. The Object Sizes are based on the percentage of bytes per page type described in [6], combined with an anecdotal representation of a web page in each category. The Composition of the Web Page Object Set attempts to generalize the various Web Page Object Media Types of real-world objects.

In our model, we have captured this range of sizes with three categories: *small* ( $\Theta_{\text{small}}$ ), *medium* ( $\Theta_{\text{medium}}$ ), and *large* ( $\Theta_{\text{large}}$ ). We assume that these Object Sizes are fixed in our model with  $\Theta_{\text{small}} = 1$ , and  $\Theta_{\text{large}} = 100$  ( $\Theta_{\text{medium}}$  will be discussed in the following paragraph). As we are only interested in the relative effect of



**Figure 6.** Visualization of the composition of the web page set.

object sizes, we do not attempt to correlate them to bytes or octets. Thus, we assume the small object size ( $\Theta_{\text{small}} = 1$ ) to be the base unit for all measures of data size in our model.

The value for the Medium Object Size ( $\Theta_{\text{medium}}$ ), was based on previous research [21]. The authors provided a detailed analysis of document sizes which we scaled to our small and large Object Sizes (between 1 and 100) and sorted from smallest to largest. By choosing the median of the document sizes based on the results from [21] as our *Medium Object Size*, we approximate the shape of the graph. We can therefore set  $\Theta_{\text{medium}} = 15$ .

Web page categories are a key feature of our model, as they allow us to characterize the effect that different compositions of web pages have on MRT. Thus, the size per web page in each category should be sufficiently different from those of other categories as to impact the MRT when the Ratio of Web Pages per Category varies. While examining web traffic patterns in [6], the authors modelled web pages as being *short*, *medium*, or *long* in terms of total page load times. They observed that *medium* pages took 3 times longer than *short* pages, and *long* pages took 6 times longer than *short* pages.

We apply their observations of *short*, *medium*, and *long* pages to our web page categories (*article*, *media*, *mosaic*) in how they affect MRT. This provides a basis for us to establish our differences in mean web page size between categories. However, to exaggerate this effect we, used higher multipliers (4 and 8) than those observed in [6].

Thus, using  $W_{\text{article}}$  as the basis, we set the approximate web page size per category to be:

$$\begin{aligned}
 W_{\text{mosaic}} &\approx 4 W_{\text{article}} \\
 W_{\text{media}} &\approx 8 W_{\text{article}}
 \end{aligned}
 \tag{3}$$

Again, the absolute value of  $W_{\text{article}}$  is not important as again we are only concerned with the relative size differences between the web page categories.

The final task in establishing the Composition of the Web Page Object Set is to determine the Number of Objects that a web page contains. For this, Equation (3) is used. The Number of Objects is an important consideration in areas such as object-level web caching [4].

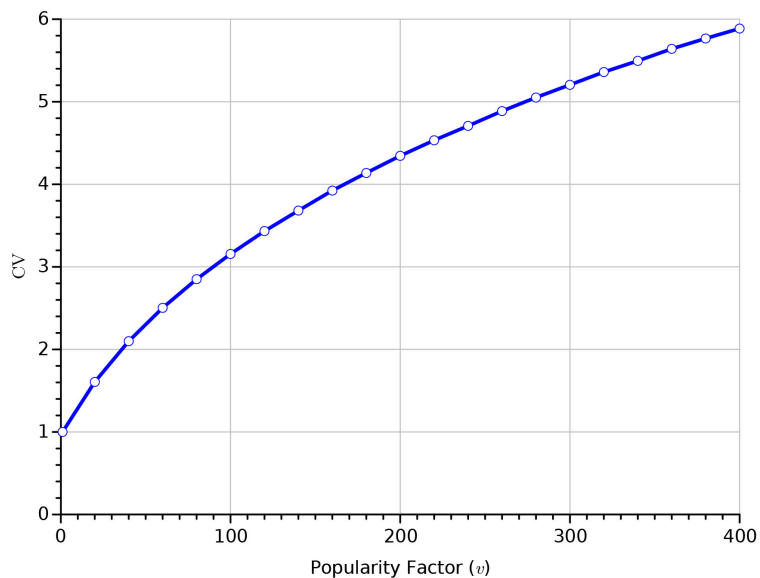
To accomplish this, we introduce one additional assumption that the average Number of Objects per web page is approximately 100 [22]. With these criteria in mind, and influenced by [6], we arrive at Table 2 which shows the Number of Web Page Objects ( $Q_i$ ) and the web page size ( $W_i$ ) for each web page based on their category.

The number of objects across our three categories (80, 138, and 63) has an average of 94, which we feel is reasonable, as it is near our goal of 100. We will use the values from Table 2 for the object composition for all remaining experiments.

#### 4.2. Coefficient of Variation of Web Page Request Interarrival Time

The *coefficient of variation of web page request interarrival time* (CV) is a metric we use for characterizing the variability of the web page request process (coefficient of variations greater than three are common [3]). The parameters we use to control the coefficient of variation are: the *number of potentially popular web pages* ( $M_0$ ), the *popularity transition rates* ( $\gamma_1$  and  $\gamma_2$ ), and the *popularity factor* ( $\nu$ ). For the number of potentially popular web pages ( $M_0$ ), we assume  $M_0 = 0.1M$  (10% of files are potentially popular) [3].

To determine the popularity factor ( $\nu$ ), we examined the effect that  $\nu$  has on the coefficient of variation; these results are shown in Figure 7. We can see from this graph that as  $\nu$  increases, so does CV, with little variation. Eventually, CV begins to level off with larger values of  $\nu$ , which is to be expected.



**Figure 7.** The effect of popularity factor ( $\nu$ ) on CV.  $K = 5$ ,  $M = 10,000$ ,  $M_0 = 1000$ ,  $\gamma_1 = \gamma_2 = 200,000$ ,  $\rho \approx 87\%$ , web page category base scenario.

**Table 2.** Number of web page objects and web page size for each web page category.

Category	$Q_i$	$W_i$
Article	80	80
Mosaic	138	320
Media	63	640

Continuing with our examination of the model parameters, we now move onto investigate the impact that coefficient of variation has on MRT (see **Figure 8**). This graph shows that MRT seems to gradually increase as CV increases. The variation in MRT also increases with CV as can be seen in the slight increase in the confidence intervals. After  $CV = 4$ , there appears to be a leveling off which is seen both in the MRT as well as in the confidence intervals. In experiments going forward, we will set  $v = 90$  to keep  $CV \approx 3$ .

### 4.3. System Load

We define system load ( $\rho$ ) to be the mean utilization over all servers in the system. It is influenced by several parameters, most notably: number of users ( $N$ ), mean web page size for all requests ( $W$ ), number of servers ( $K$ ), and coefficient of variation (CV).

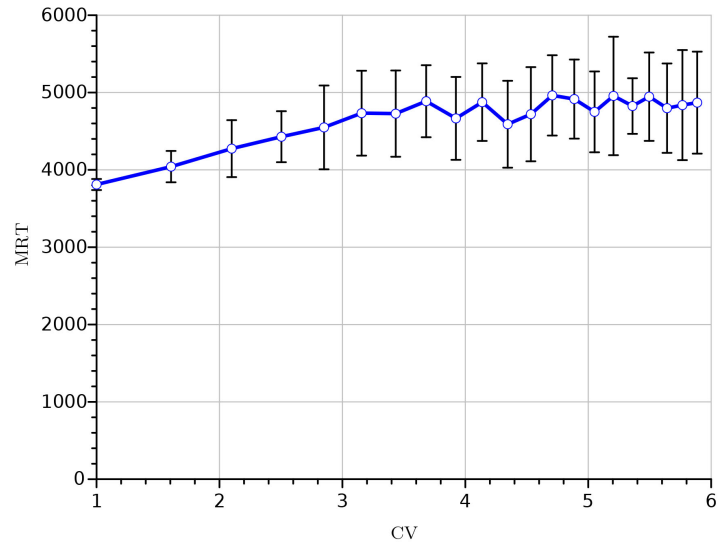
In **Figure 9**, we examine how the system load varies with the number of users and the number of servers (where  $K = \{1, 5, 10\}$ ). We see that for all server scenarios, system load ( $\rho$ ) increases with  $N$ . In addition, as the  $K$  increases,  $N$  must increase to achieve equivalent system load ( $\rho$ ). In all three scenarios, there is a distinct leveling off when the number of users ( $N$ ) increase to the point where the systems become saturated ( $\rho = 100\%$ ) [23].

We complete our investigation of system load in **Figure 10**, where we observe the effect of varying the number of users ( $N$ ) has on MRT. Lower values of  $N$  seem to have a minimal effect on MRT, however, as  $N$  continues to increase, the effect on MRT transitions to a higher rate of change. This transition corresponds to approximately the same points where the change in slope levels off in **Figure 9**. The rate of increase in MRT grows linearly until system load approaches 100%. It is also evident from **Figure 10** that increasing the number of servers ( $K$ ) reduces the MRT as the number of users ( $N$ ) increases.

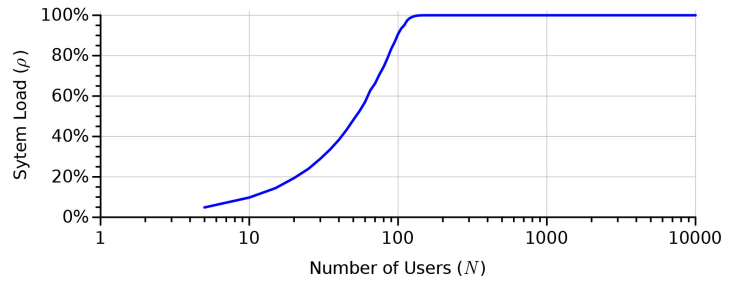
### 4.4. Varying Web Page Category

As was mentioned in Section 4.1, our analysis of web page categories is a key feature of our model as it characterizes the effect that different compositions of web pages have on MRT. In this section, we present results of varying the *Ratio of Web Pages per Web Page Category* on MRT.

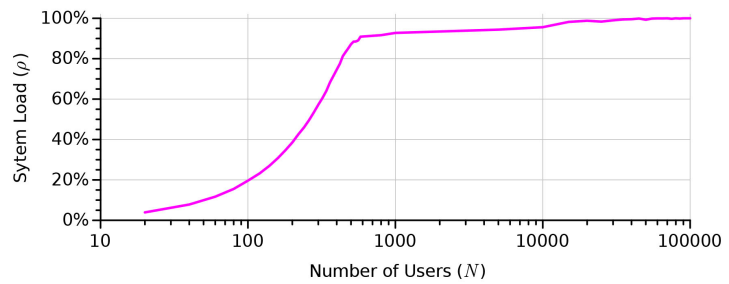
**Table 3** shows the thirty-six *Web Page Category Ratio Test Scenarios* that we developed, which are made up of permutations of Ratio of Web Pages per Web Page Category in increments of 10%, starting at 10%.



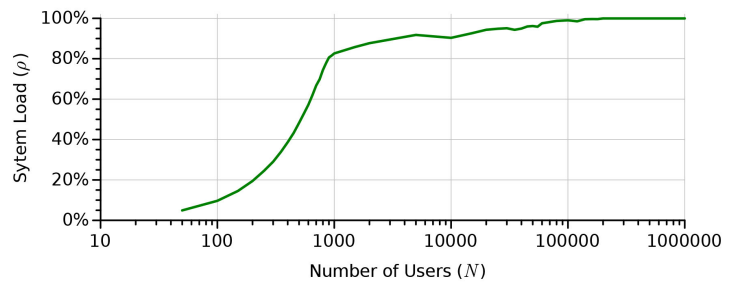
**Figure 8.** The effect of CV on MRT.  $K = 5$ ,  $M = 10,000$ ,  $M_0 = 1000$ ,  $\gamma_1 = \gamma_2 = 200,000$ ,  $\rho \approx 87\%$ , web page category base scenario.



(a)  $K = 1$



(b)  $K = 5$

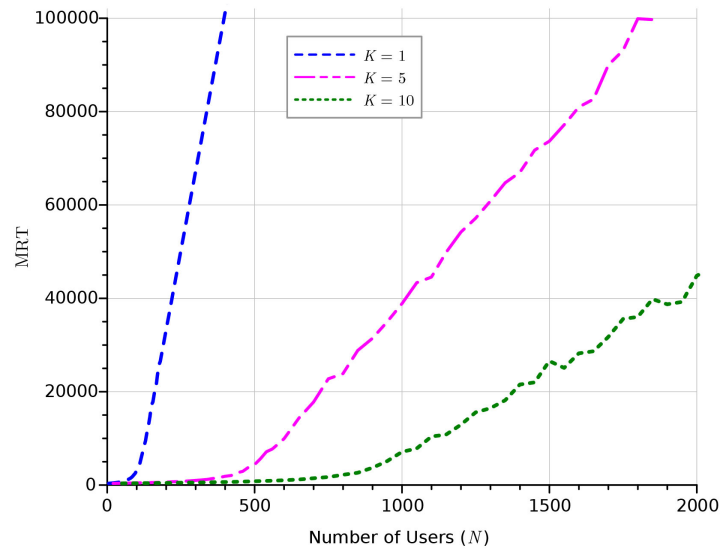


(c)  $K = 10$

**Figure 9.** Varying system load with number of users.  $M = 10,000$ ,  $M_0 = 1000$ ,  $CV \approx 3$ , web page category base scenario.

**Table 3.** Web page category ratio test scenarios.

Scenario	$k_{\text{article}}$	$k_{\text{mosaic}}$	$k_{\text{media}}$
1	10	10	80
2	10	20	70
3	10	30	60
4	10	40	50
5	10	50	40
6	10	60	30
7	10	70	20
8	10	80	10
9	20	10	70
10	20	20	60
11	20	30	50
12	20	40	40
13	20	50	30
14	20	60	20
15	20	70	10
16	30	10	60
17	30	20	50
18	30	30	40
19	30	40	30
20	30	50	20
21	30	60	10
22	40	10	50
23	40	20	40
24	40	30	30
25	40	40	20
26	40	50	10
27	50	10	40
28	50	20	30
29	50	30	20
30	50	40	10
31	60	10	30
32	60	20	20
33	60	30	10
34	70	10	20
35	70	20	10
36	80	10	10



**Figure 10.** Effect of number of users on MRT.  $M = 10,000$ ,  $M_0 = 1000$ ,  $K = 5$ ,  $CV \approx 3$ , web page category base scenario.

**Figure 11** presents a graph of the results of the various test scenarios on Mean Web Page Size for All Requests ( $\bar{W}$ ). We can see a general trend where, with low ratios of Article Web Pages,  $\bar{W}$  tends to be high, being dominated by the larger Mosaic and Media Web Pages. Conversely, with higher ratios of Article Web Pages, Mosaic and media Web Pages tend to contribute less to  $\bar{W}$ . This makes sense since the relative difference in web page size has Mosaic and Media Web Pages 4 and 8 times larger than Article Web Pages.

The Mean Web Page Size for our Base Scenario is shown also noted in **Figure 11** and has a value of 344.7. This puts it roughly in the middle of the minimum and maximum Mean Web Page Sizes (158 and 554). This confirms our goal to have the Mean Web Page Size for our Base Scenario be near the middle range of  $\bar{W}$ .

Next, we examine how the Mean Web Page Size affects MRT and system load. These results are shown in **Figure 12**. We can observe that MRT increases non-linearly as  $\bar{W}$  increases, and at higher values of  $\bar{W}$ ,  $\rho$  flattens out around 90%. This is in line with the results we saw in Section 4.3. The system load for our Base Scenario is 85,3%. This is not a coincidence, as we have chosen parameters so that Base Scenario results in  $\rho \approx 85\%$ .

Of particular interest for these experiments is that we can now establish three Web Page Category Ratio Test Scenarios that could be used as representative cases when incorporating our User Request Object-Level framework model into the study of other distributed web applications. Along with our Base Scenario (established in Section 4.1), we have added Low and High scenarios. These three Test Scenarios are summarized in **Table 4**.

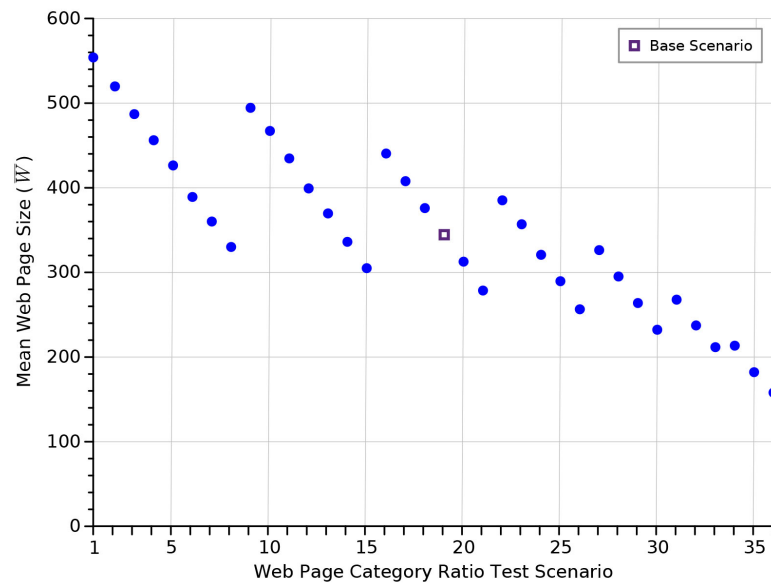
We now examine the effect that these Test Scenarios have on MRT (results shown in **Figure 13**). We can observe that MRT increases as the mean web page size ( $\bar{W}$ ) increases. This is reasonable since  $\bar{W}$  is proportional to request service time. The relationship between  $\bar{W}$  and MRT is however, non-linear, but we can



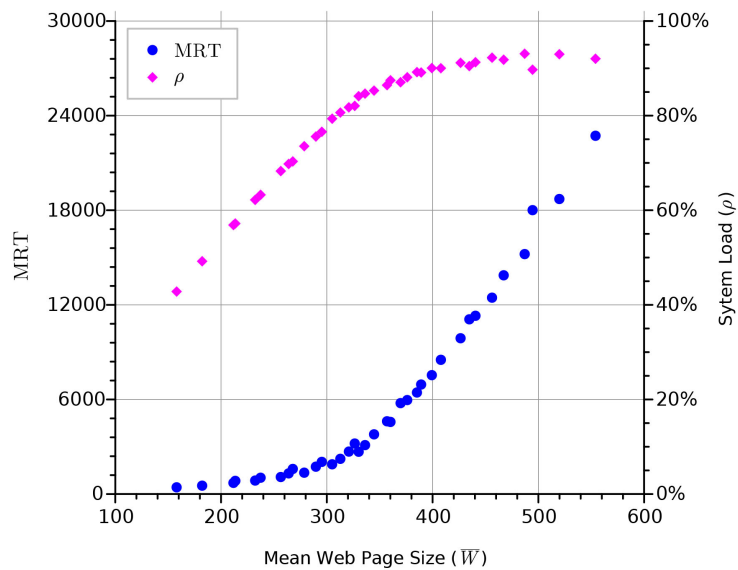
observe that the difference between the values of the MRT and  $\bar{W}$  are pronounced. One of our goals in Section 4.1 was that each web page category be different enough to significantly impact MRT which we have certainly established.

**Table 4.** Summary of web page category ratio test scenarios.

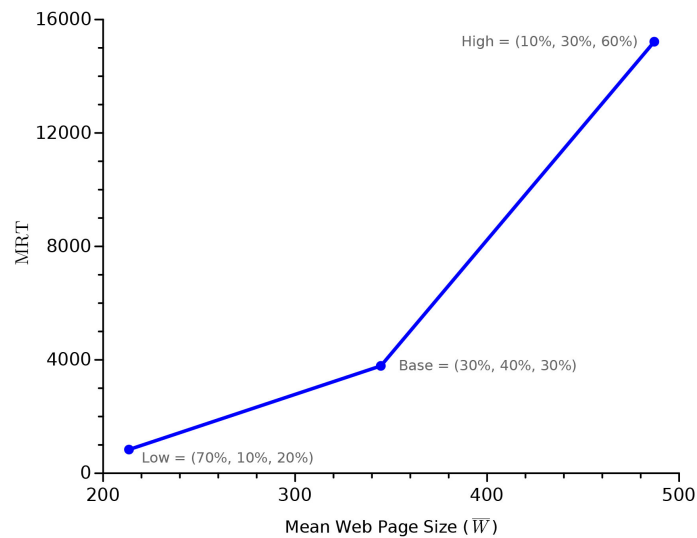
Scenario	$k_{\text{article}}$	$k_{\text{mosaic}}$	$k_{\text{media}}$	Measured $\bar{W}$
Low	70	10	20	213.5
Base	30	40	30	344.7
High	10	30	60	487.1



**Figure 11.** Effect of varying the ratio of web pages per web page category on the mean web page size for all requests.



**Figure 12.** The effect that mean web page size has on MRT and system load.  $M = 10,000$ ,  $M_0 = 1000$ ,  $CV \approx 3$ ,  $K = 5$ ,  $N = 480$ ,  $z = 35,000$ .



**Figure 13.** The effect of our final web page category ratio test scenario on MRT.  $K = 5$ ,  $N =$ ,  $z = 35,000$ ,  $CV \approx 3$ ,  $\rho_{Low} \approx 57\%$ ,  $\rho_{Base} \approx 85\%$ ,  $\rho_{High} \approx 93\%$ .

#### 4.5. Conclusions

In this paper, we presented a User Request Object Level Framework Model which is composed of three sub-models: *User-Request Model*, *Server Model*, and *Web Page Model*. Of particular interest is the *Web Page Model*, which categorizes the composition of web pages at the object level and represents the presentation of the web page from the user's perspective. We used our framework model to establish and evaluate system parameters and demonstrated that there is a relationship between MRT and the composition of the Web Page Set. Based on those results, we were able to develop three Web Page Category Ratio Test Scenarios (Base, Low, and High) that can be used to investigate distributed applications.

One of the main goals of our research was to develop a framework model that can be incorporated into other research. The obvious choice from this work would be to apply the framework (more specifically, the three Web Page Category Ratio Test Scenarios) to a web-caching environment which is set up with web pages composed of objects and a user community generating requests. Another possible application is the *Internet of Things* (IoT). The IoT is a concept of interconnecting small devices and other systems for exchanging data. Although the types of IoT devices span a large range of computing capabilities and data requirements, the overall environment can be envisioned to consist of a user population interacting with devices that contain objects that are requested. An IoT environment would differ from the one examined in this paper as it would have a higher ratio of web pages with respect to our developed categories, and a page would be made up of a small number of smaller number of objects that had low catchability.

As for the framework itself, it could be extended to include other object types.

Our model used six web page object media types that generalize the many that are available in the WWW. An example of this is the text object type, which is intended to be representative of HTML and XML objects. These objects, when compared to one another, have been shown to have different request rates and represent different request size proportions. These can easily be thought of as separate types. In addition, the object type, octet, was left out of our model, and it represents a moderate amount of data transferred on the WWW, especially for larger web pages. Octet objects are often related to video watching and large files. It would be interesting to see the effect of modelling these, and other object types.

### Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

### References

- [1] Freed, N. and Kucherawy, M. (2017) Media Types. <https://www.iana.org/assignments/media-types/media-types.xhtml>
- [2] Ali, W., Shamsuddin, S.M. and Ismail, A.S. (2011) A Survey of Web Caching and Prefetching. *International Journal of Advances in Soft Computing and Its Applications*, **3**, 18-44.
- [3] Hurley, R. and Plumley, B. (2018) Comparison of Sender and Receiver-Initiated Load Balancing in a Distributed Web Caching System. *Proceeding of the 33rd International Conference on Computers and Their Applications (CATA2018)*, Las Vegas, 19-21 March 2018, 78-84.
- [4] Sturgeon, R. (2022) Modelling Request Access Patterns for Information on the World Wide Web. Master's Thesis, Trent University, Ontario.
- [5] Chen, J.J. and Cheng, W.Q. (2016) Analysis of Web Traffic Based on Http Protocol. *2016 24th International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, Split, 22-24 September 2016, 1-5. <https://doi.org/10.1109/SOFTCOM.2016.7772120>
- [6] Ihm, S. (2011) Understanding and Improving Modern Web Traffic Caching. Ph.D. Thesis, Princeton University, Princeton. <https://doi.org/10.1145/2068816.2068845>
- [7] Newton, B., Jeffay, K. and Aikat, J. (2013) The Continued Evolution of Web Traffic. *2013 IEEE 21st International Symposium on Modelling, Analysis and Simulation of Computer and Telecommunication Systems*, San Francisco, 14-16 August 2013, 80-89. <https://doi.org/10.1109/MASCOTS.2013.16>
- [8] Barford, P. and Crovella, M. (1998) Generating Representative Web Workloads for Network and Server Performance Evaluation. *ACM SIGMETRICS Performance Evaluation Review*, **26**, 151-160. <https://doi.org/10.1145/277858.277897>
- [9] El Abdouni Khayari, R., Musovic, A., Lehmann, A. and Fellinger, P. (2009) A Workload Based Adaptive Scheduling Algorithm for Web Server. *Proceedings of the 2009 Spring Simulation Multiconference*, San Diego, 22-27 March 2009, 1-8.
- [10] Ihm, S. and Pai, V.S. (2011) Towards Understanding Modern Web Traffic. *Proceeding of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference*, Berlin, 2-4 November 2011, 295-312. <https://doi.org/10.1145/2068816.2068845>

- [11] Evans, D. (2011) The Internet of Things: How the Next Evolution of the Internet Is Changing Everything. Technical Report, Cisco IBSG.
- [12] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P. and Berners-Lee, T. (1999) RFC 2616, Hypertext Transfer Protocol—HTTP/1.1. <https://doi.org/10.17487/rfc2616>
- [13] Hurley, R.T. and Li, B.Y. (2008) Effects of Dynamic Content on Web Caching. *Proceeding of the ISCA 21st International Conference on Parallel and Distributed Computing and Communication Systems (PDCCS'08)*, New Orleans, 24-26 September 2008, 165-170.
- [14] Gangemi, A. and Presutti, V. (2006) Towards an Owl Ontology for Identity on the Web. *Proceedings of the 3rd Italian Semantic Web Workshop*, Volume 201, Pisa, 18-20 December 2006.
- [15] Tanenbaum, A.S. and van Steen, M. (2006) *Distributed Systems: Principles and Paradigms*. 2 Edition, Prentice-Hall, Upper Saddle River.
- [16] Stallings, W. (2007) *Data and Computer Communications*. 8th Edition, Pearson Prentice Hall, Old Bridge.
- [17] Conklin, J. (1987) A Survey of Hypertext. Technical Report, Microelectronics and Computer Technology Corporation.
- [18] Banks, J., Carson, J., Nelson, B. and Nicol, D. (2005) *Discrete-Event System Simulation*. 4th Edition, Pearson Education Inc., London.
- [19] Hurley, R.T. and Li, B.Y. (2008) A Performance Investigation of Web Caching Architectures. *Proceeding of the Canadian Conference on Computer Science and Software Engineering*, Montreal, 12-13 May 2008, 184-188.
- [20] Bodnarchuk, R.R. and Bunt, R.B. (1991) A Synthetic Workload Model for a Distributed System File Server. 1991 *ACM SIGMETRICS*, San Diego, 21-24 May 1991, 50-59. <https://doi.org/10.1145/107971.107978>
- [21] González-Canete, F., Casilari-Pérez, E. and Trivino, A. (2007) Characterizing Document Types to Evaluate Web Cache Replacement Policies. *Fourth European Conference on Universal Multiservice Networks (ECUMN'07)*, Toulouse, 14-16 February 2007, 3-11. <https://doi.org/10.1109/ECUMN.2007.11>
- [22] WebSiteOptimization.com (2012) Average Number of Web Page Objects Breaks 100—Web Page Object Statistics and Survey Trends. <http://www.websiteoptimization.com/speed/tweak/average-number-web-objects>
- [23] Kleinrock, L. (1975) *Queueing Systems Volume I: Theory*. Wiley, New York.