GOCE DELCEV UNIVERSITY - STIP
FACULTY OF COMPUTER SCIENCE

ISSN 2545-4803 on line

BALKAN JOURNAL OF APPLIED MATHEMATICS AND INFORMATICS (BJAMI)

YEAR 2020

VOLUME III, Number 2

GOCE DELCEV UNIVERSITY - STIP, REPUBLIC OF NORTH MACEDONIA
FACULTY OF COMPUTER SCIENCE

ISSN 2545-4803 on line

BALKAN JOURNAL OF APPLIED MATHEMATICS AND INFORMATICS

(BJAMI)

YEAR 2020 VOLUME III, Number 2

AIMS AND SCOPE:
BJAMI publishes original research articles in the areas of applied mathematics and informatics.

Topics:
1. Computer science;
2. Computer and software engineering;
3. Information technology;
4. Computer security;
5. Electrical engineering;
6. Telecommunication;
7. Mathematics and its applications;
8. Articles of interdisciplinary of computer and information sciences with education, economics, environmental, health, and engineering.

Managing editor
Biljana Zlatanovska Ph.D.

Editor in chief
Zoran Zdravev Ph.D.

Lectoure
Snezana Kirova

Technical editor
Sanja Gacov

Address of the editorial office
Goce Delcev University – Stip
Faculty of philology
Krste Misirkov 10-A
PO box 201, 2000 Štip,
Republic of North Macedonia

BALKAN JOURNAL
OF APPLIED MATHEMATICS AND INFORMATICS (BJAMI), Vol 3

ISSN 2545-4803 on line
Vol. 3, No. 2, Year 2020

EDITORIAL BOARD

CONTENT

Mirjana Kocaleva and Vlado Gicev
A METHODOLOGY OF DETERMINING THE NATURAL
FREQUENCIES OF LOW-RISE BUILDINGS 7

Boro M. Piperevski and Biljana Zlatanovska
ABOUT ONE B.S. POPOV'S RESULT 15

Lyubomir Filipov and Zlatko Varbanov
STATIC GENERATION OF WEBSITES –
POSITIVE AND NEGATIVE ASPECTS 25

Goce Stefanov, Vasilija Sarac, Maja Kukuseva Paneva
SINE INVERTER SYSTEM BASED ON SPECIAL CIRCUIT 31

Blagica Doneva, Marjan Delipetrev, Gjorgi Dimov
BASIC PRINCIPLES OF GEOMETRIC SEISMICS 43

Normin Okićić and Amra Rekić-Vuković
ON KURATOWSKI MEASURE OF NONCOMPATNESS IN R^2
WITH RIVER METRIC 59

Shinta Puspasari and Henky Honggo
EIGEN SPACE BASED CHARACTER RECOGNITION APPLICATION
FOR LEARNING CHINESE HANDWRITING IN PALEMBANG 69

UDC: 004.774-026.11

STATIC GENERATION OF WEBSITES – POSITIVE AND NEGATIVE ASPECTS

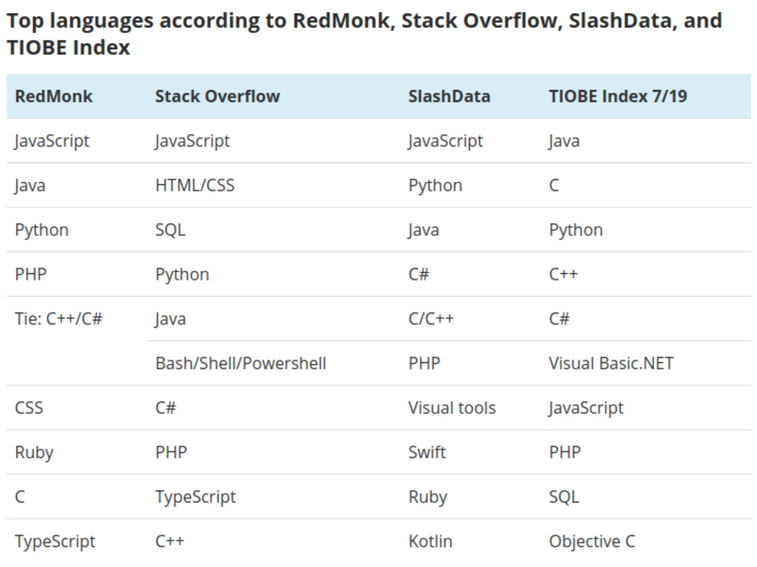
LYUBOMIR FILIPOV AND ZLATKO VARBANOV

Abstract. Static generation is a technique used in web content delivery, fetching of content, and other various aspects. In this paper, static generation will be investigated in the aspects of high traffic web systems whose main purpose is to show content. Basic principles will be covered by pointing out the positive and negative aspects of the static generation.

1. **Introduction**

Widespread development of web pages is related to the HTTP protocol. The Hypertext Transfer Protocol (HTTP) is an application-level protocol for distributed, collaborative, hypermedia information systems. HTTP has been in use by the World-Wide Web global information initiative since 1990 [1]. The following step is having the HTML standard accepted in 1996. Dynamic web content is marked with the server-side scripting which has first references in 1998 in Server-Side JavaScript Guide [2]. Dynamic websites are related to scripting languages. Currently based on a survey from various origins, we have the following chart of programming, scripting, and markup languages.

Table 1. *Top languages chart*[1]



Top languages according to RedMonk, Stack Overflow, SlashData, and TIOBE Index

RedMonk	Stack Overflow	SlashData	TIOBE Index 7/19
JavaScript	JavaScript	JavaScript	Java
Java	HTML/CSS	Python	C
Python	SQL	Java	Python
PHP	Python	C#	C++
Tie: C++/C#	Java	C/C++	C#
	Bash/Shell/Powershell	PHP	Visual Basic.NET
CSS	C#	Visual tools	JavaScript
Ruby	PHP	Swift	PHP
C	TypeScript	Ruby	SQL
TypeScript	C++	Kotlin	Objective C

The data shown on the diagram is from 2019, which points us to the following languages: JavaScript (NodeJS and all other frameworks), PHP, C#, Java, and Python as the most common scripting languages responsible for the delivery of dynamic website content. On the other side, the increasing amount of internet users should be measured. Based on the data in 1995, 16 million users had access to the internet; compared to 2019, these figures now show 4,383 million users[2].

[1] Diagram with data - https://blog.newrelic.com/technology/most-popular-programming-languages-of-2019/
[2] Internet Growth Statistics - https://www.internetworldstats.com/emarketing.htm

Based on the two facts stated above, nowadays it is common practice to have systems that have to support multiple users at the same time.

High traffic systems usually rely on fast load and a good look and feel for the end clients. On the market, these are connected to delivering complex systems that could not run on a simple server, but it requires way more resources.

The common practice now is to generate static websites based on a dynamic one. All the content is served by CDN (Content Delivery Network) and it is distributed across different endpoints.

To cut off costs for hosting purposes and deliver a scalable product, the static generation technique is used. Nowadays there are various content management systems (CMS) that are used across the globe. Some of them are paid with a license, others are open-source but they share the same purpose. Content management generally relates to storing and administering various types of data files [4]. These systems cover access to certain data, versioning, workflow moderation, files management, and user/editor management. The editors have the power to present the content in almost any form they like to the end user.

Potential issues with editorial work are:

1. End user seeing content which is not supposed to be published;
2. End user experiencing broken links to content or files while the editor is working;
3. Different user experience based on whether someone is logged or not;
4. Locking content only to the author;
5. Due to high traffic - the editor unable to operate with the CMS.

All these issues have a common solution related to static generation.

2. Static generation techniques

There are basically two techniques:

- Crawl website and generate static using an external tool (Wget);
- Using other modules or custom code that will generate a static version.

Using other modules or custom code will have to perform the same operations that Wget is doing. GNU Wget is a free utility for non-interactive download of files from the Web. It supports HTTP, HTTPS, and FTP protocols, as well as retrieval through HTTP proxies. In this case, “non-interactive” means that it can work in the background, while the user is not logged on. This allows it to start a retrieval and disconnect from the system, letting Wget finish the work. By contrast, most of the Web browsers require constant user’s presence, which can be a great obstacle when transferring a lot of data.

Wget can follow links in HTML, XHTML, and CSS pages to create local versions of remote web sites, fully recreating the directory structure of the original site. This is sometimes referred to as "recursive downloading." While doing that, Wget respects the Robot Exclusion Standard (/robots.txt). Wget can be instructed to convert the links in downloaded files to point at the local files, for offline viewing [3]. Most of the features are fully configurable.

Wget has some drawbacks mainly related to responsive images and not reading properly all HTML tags and attributes. A typical example of this behavior is the srcset attribute which is not read at all in certain Wget versions.

2.1. **Process of static generation**

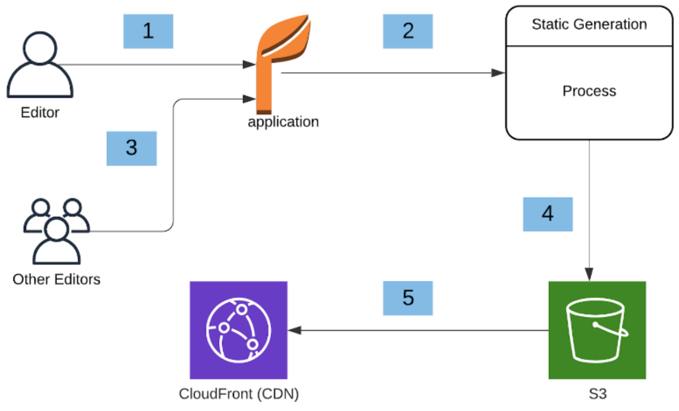


Figure 1. *Process of static generation*

The steps of this process are the following:

1. Editor initiates static generation to the application;
2. Application locks edit of data, prepares a list of any custom URL's that should be crawled and generated static, and prepares a file for the error page. It starts the static generation process, which is crawling the system and saving all the data in a folder-based structure;
3. All other editors are banned from editing the system while static generation is running to avoid bad data and any failures;
4. Once all resources are downloaded to a folder, an error page file is generated and all extra resources are prepared - sync of folder data is performed to the host of static files (S3);
5. Once all the data has been transferred to the bucket, the CDN cache is invalidated so the end users could see the newly published content.

All the steps above are using the AWS (Amazon Web Services) stack, but the process could be duplicated in various hosting providers. The main steps are locking edit of content, generating static version (Wget or custom logic), making sure that the error page is set, then moving all the files to a static host, and setting up CDN if any changes are required.

2.2. **Drawbacks**

Not all systems could become static. Dynamic parts should be handled by microservices and API calls. Dynamic parts are listing data from Databases or Elastic Services, form submissions that require extra validation of business logic. A good approach might be to have microservices responsible for the dynamic parts. If the microservice is down, the system will still be functional; only that part will not be available.

When generating static versions, the following should be taken into consideration:

1. Speed of generation;
2. Access to assets (is any moving of files required?);
3. Cache invalidation.

Based on the content in CMS, there might be a case where static generation is taking over an hour, this is something that has to be communicated with the business. Access to assets (direct links to pdf for example) is something that needs to be organized, because if there is an asset that is not linked anywhere in the site, then the static generation will not move it to the static version. Cache invalidation is always an issue when using CDN; proper cache invalidation must be performed at any cost after static generation.

3. **Conclusions**

Positive points:

1. There is no issue with server time;
2. CDN guarantees that access is the same across the world; performance-wise the first call will be the slowest one, then, based on good optimization and reuse of assets, most of the data will be cached on the client-side. CDNs provide cache on the server-side as well which, in combination with lazy loading and absence of dynamic data, guarantees speed.
3. Hosting costs are limited to a minimum (simple static bucket in Amazon could serve as a server); combination of CDN and static bucket relies on the infrastructure behind the provider (in this example Amazon), S3 bucket is available or not, and there is no limit on the number of connections to it. Usually providers like Amazon distribute the files across several servers which is not visible for the end user but internally it scales in order to serve the demand.
4. There is a strong difference between what is published and what is not from an editorial point of view;
5. It could serve high traffic and no downtime is expected;
6. Absence of dynamic parts increases the security - one could not inject something dynamic; an attacker could not alter the scripts because the CDN providers have a strong policy regarding about who the owner of the content is and a limit of the available headers. Any dynamic parts could be secured by checks based on headers, origin, and tokens that could be generated on every call;

Negative points:

1. Complex architecture;
2. Harder to maintain (issues with Wget or any other static content);
3. Potential cross-origin resource sharing issues;
4. Need for microservices for dynamic parts (search and filters, form submissions);
5. Potential issues with tracking (marketing-related);

Based on all the positive and negative points we could conclude that static generation is a good technique that could be implemented only in certain business cases.

References

[1] *Fielding, R., Gettys, J., Mogul, J., Frystyk, H., and Berners-Lee, T.* (1999). Hypertext Transfer Protocol -- HTTP/1.1, RFC 2068, DOI 10.17487/RFC2068, available online at: https://www.w3.org/Protocols/HTTP/1.1/rfc2616.pdf

[2] *Husted, R., Kuslich, J.* (1999). Server-Side JavaScript(TM): Developing Integrated Web Applications Paperback, Addison-Wesley, book.

[3] *Niksic, H.* (2015). GNU Wget: The non-interactive download utility, Samurai Media Limited, book.

[4] *Parnell, T., Uzquiano, M., Royston, S.* (2000). Open Invention Network LLC 2000, Classification based content management system, US6647396B2, patent, available online at https://portal.unifiedpatents.com/patents/patent/US-6647396-B2.

Lyubomir Filipov
University of Veliko Tarnovo,
Faculty of Mathematics and Informatics,
Address: 2 T.Tarnovski str., Veliko Tarnovo
Country: Bulgaria
E-mail address: lyubomir.g.1991@gmail.com

Zlatko Varbanov
University of Veliko Tarnovo,
Faculty of Mathematics and Informatics,
Address: 2 T.Tarnovski str., Veliko Tarnovo
Country: Bulgaria
E-mail address: zl.varbanov@ts.uni-vt.bg