December 26, 2018

Simulating web page loading performance

Patrick Hulce

Simulating web page loading performance

ABSTRACT

Determining the load time for a web page in any specific environment can be challenging and slow owing to the diversity of devices and network types. This disclosure provides techniques to estimate the loading time of a web page in a given environment via simulation of a dependency graph constructed by tracking the page loading process in any available environment. The simulated traversal activates a node only when all of its dependencies in the graph have completed. The time taken for node completion in the simulation is varied depending on the corresponding estimate for the target environment in which the web page would be loaded.

KEYWORDS

- Page load time
- Page load performance
- Performance simulation
- Performance analysis
- Browser
- Device environment
- Environment-specific search results

BACKGROUND

The time taken to load and display a web page varies based on various environmental factors, such as the capabilities of the requesting device and the quality and bandwidth of the network connection. Having a sense of the load time of a web page can be important for providing a desirable User eXperience (UX). However, determining the load time for a web page

in any specific environment may require the use of repeated observations and heuristics. Employing such techniques can be challenging and slow owing to the diversity of devices and network types.

DESCRIPTION

This disclosure provides techniques to estimate the loading time of a web page in a given environment, e.g., characterized by a device type, network type, etc. The target page is loaded in any available environment and the page loading process is tracked in detail. The tracked aspects include information collected from a web browser that loads the page about the resources necessary to load the page, resources added by other resources, tasks executed by the device processor, etc.

The collected information is utilized to construct a dependency graph that depicts all relationships between all resources with each node representing a specific resource involved in the loading of the web page. The dependency graph is further broken down by extracting multiple subgraphs that represent key events in the page loading process representing the performance of the various resources at that moment. By combining these subgraphs with the technical specifications of the various standard protocols, such as Transmission Control Protocol (TCP), HyperText Transfer Protocol (HTTP), Secure Sockets Layer (SSL), etc., the entire dependency graph can be simulated by traversing each node. The simulated traversal activates a node only when all of its dependencies in the graph have completed. The time taken for node completion in the simulation is varied depending on the corresponding estimate for the target environment in which the web page would be loaded.

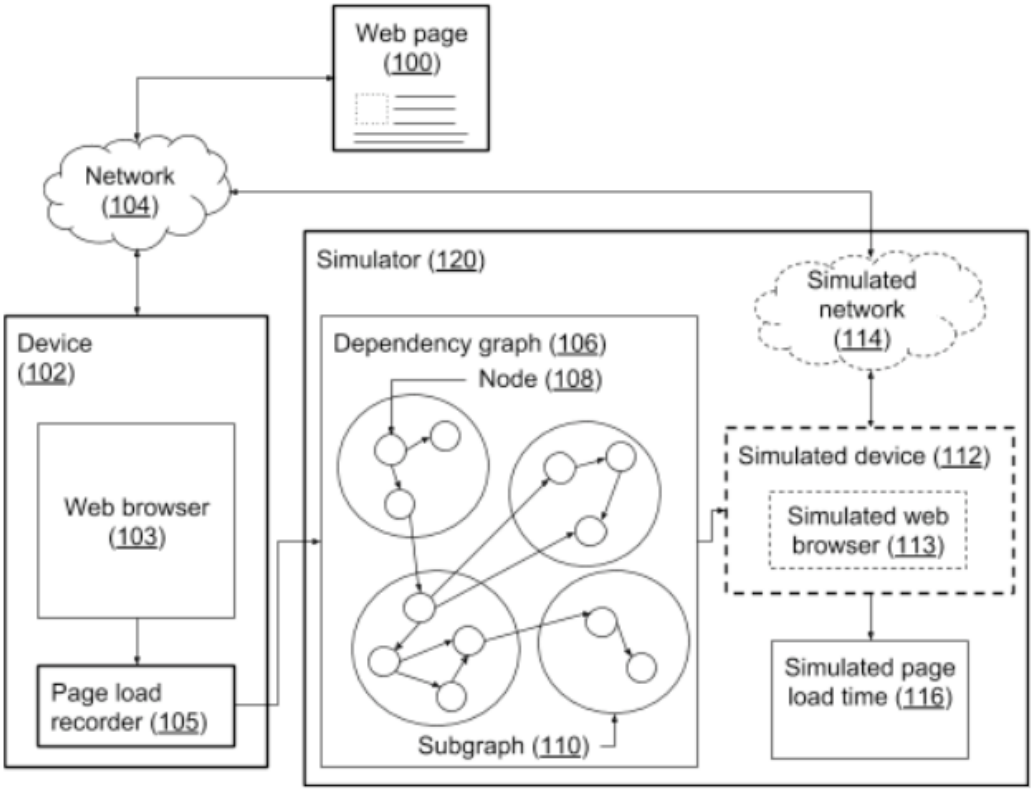

Fig. 1: Recording web page loading on one device and simulating it for another device

Fig. 1 shows a web page (100) being requested and loaded by a web browser (103) on a device (102) via a network connection (104). The loading process is tracked by a page load recorder (105) and is used to generate a dependency graph (106) that can be utilized to simulate page loads. Nodes (108) of the dependency graph represent the various resources, such as scripts, images, text, videos, etc., involved in the web page loading process.

The dependency graph is composed of several subgraphs (110) that represent the performance of the various resources at several key moments during the page load process. The dependency graph with its subgraphs is then utilized by a simulator (120) to simulate the loading of the same web page in a web browser (113) on a simulated device (112) via a simulated network connection (114). The environment for this second access of the web page, such as the

capabilities of the device (112) and the bandwidth of the network connection (114) may differ from the original environment under which the page was accessed to create the dependency graph and its subgraphs. When simulating the loading of the page, the dependency graph is traversed from the starting node activating subsequent dependent nodes upon its completion. Completion times for the nodes are determined based on the corresponding estimates for environmental conditions of the access by the simulated device (112) via the simulated network (114). The simulation process yields an estimated load time (116) for the web page when accessed by the device (112).

The approach presented in this disclosure enables estimating the web page loading performance in a given environment based only on a single recording of the web page loading process in any available environment. Owing to the use of the recorded dependency graph of the actual page loading process, the simulation can provide better accuracy than heuristics as well as statistical estimation. Further, the simulation process avoids the need for a large corpus of comparison data for computing the multipliers. As an alternative to direct simulation, the process can be implemented utilizing a trained machine learning model to obtain similar results.

One practical application of the techniques of this disclosure can enhance the presentation of search results. For instance, the results of a search query may show only those pages that can be successfully loaded within a reasonable time given the constraints of the environment under which the query was issued. For example, the results for a query issued from a mobile phone would show only those pages that may be properly loaded and viewed on a smaller screen with a mobile browser and the limited bandwidth of a mobile network. The techniques of this disclosure can also be integrated within web browsers to measure various metrics for page loading that can be used by in various ways by developers as well as end users.

CONCLUSION

The approach presented in this disclosure enables estimating web page loading performance in a given environment based only on a prior recording of the same web page loading in any available environment. The recording is used to generate a dependency graph of the page loading process along with subgraphs representing key events within the process. Simulated traversal of the dependency graph activates a node only when all of its dependencies in the graph have completed. The time taken for node completion in the simulation varies depending on the corresponding estimate for the target environment in which the web page would be loaded. The techniques of this disclosure can also be integrated within a web browser and utilized to enhance search result listings.