Experimental study on round trip times of web applications

Rafi Khaled, Rui Meireles
{rkhaled, rui.meireles}@vassar.edu

Department of Computer Science, Vassar College, USA

***Abstract*—Replace by concrete abstract.**

I. Introduction

Introduction.
Figure 1 is an example figure.

II. Related work

Related work.

III. Methodology

Our goal was to observe Round Trip Time (RTT) of the request to a webserver isolated from the network time. **(Need clarification and further guidance on the goals and wording)**. Collection of data was conducted using a pure ICMP ping implementation [1] and a client-side URL transfer library [2]. ICMP ping data was collected in order to contrast it to the data collected from the client-side URL transfer library; the former of which, in contrast to the latter, does not go all the way up the networking stack to the webserver application and instead immediately replies to the client. In other words, the delay of the webserver is not captured when using ICMP ping data. Having both sources of data provided insight on the origin of the RTT delay. **(I am not sure that this paragraph is an accurate description or is too much detail for the methodology section)**

First, we wanted to calculate the RTT as measured by a calculation involving, at the highest level, a website's pre-transfer time and time to first byte (TTFB): the RTT was calculated as the pretransfer time subtracted from the TTFB, both of which were obtained with a client-side URL transfer library using Curl **(Am I being too specific about the implementation by saying we used Curl?)** [2]. To do this, simply provide the hostname of a website to a Curl Object. In our implementation, we configured the options of the library to follow any Location: header that the server sends as part of a HTTP header in a 3xx response [3] **(I am not sure exactly WHY I did this, but if it was something you suggested then some guidance would be appreciated. see [3] for details about setting FOLLOWLOCATION to 1 in Curl)**.

The next calculation we wanted to observe is the RTT on the second load of a website. This calculation has the same equation of RTT as described above, except with the appropriate second load of the pretransfer time and TTFB. In order to measure this, we took advantage of Curl allowing for reuse of Objects: one only needs to reset the relevant options identically to the first load and can then perform on the Curl Object. This process was repeated to calculate the RTT on the third load. **(Again, am I being too specific about the implementation here?)**

Finally, we used pure ICMP to get ping RTTs for every website.

The above network data; i.e. ICMP ping time and RTT of first, second, and third loads, were collected for a list of some of the most popular websites around the world at the time of writing this paper. A cron job was used to automate this collection of data to achieve a large sample size to analyze. We ran the cronjob around once every ten minutes for a week. **(We can definitely change the frequency of the cronjob and collect data if needed)**

In our analysis, we observed both aggregate data and site-specific data by graphing their Cumulative Distribution Function, as shown in Fig. 1 below.

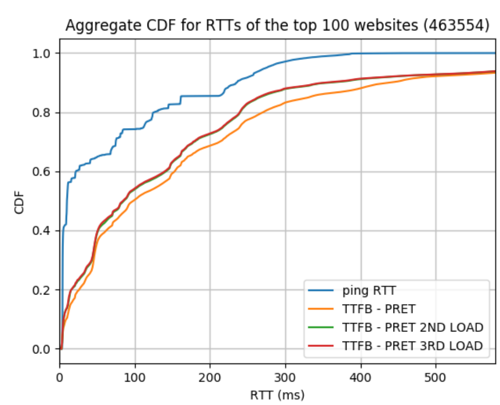

Fig. 1. Cumulative Distribution Function for aggregate RTT data

IV. Evaluation

Evaluation.

V. Conclusions

Conclusions.

References

[1] M. D. Cowles. pyping 0.0.5. [Online]. Available: https://pypi.python.org/pypi/pyping/

[2] (2016, Dec) Pycurl 7.43.0.1. [Online]. Available: http://pycurl.io/
[3] Curlopt˙followlocation explained. [Online]. Available: https://curl.haxx.se/libcurl/c/CURLOPT_FOLLOWLOCATION.html