Project Report

----How Cache Affect Page Load Time and Critical Path Using WProf

1. Introduction & Problem Description

I'm Yunke Tian (109929662). I'm taking this course together with CSE 522, and the special request for this is that I should do this course project alone.

The problem I'm studying on is "How does cache affect page load time (PLT) and critical rendering path". To be specific, I'm studying, when loading webpages, what's the behavior of PLT and Critical path with and without local http cache and DNS cache, and how specifically cache affects them.

2. Solution Methodology

To do so, below is what I've done:

a. Environment setup. So I was provided with relevant codes by Javad from his experiment using WProf. However, the aim of his experiment is different from mine, and I cannot directly use his codes. First I downloaded his codes, and install Ubuntu OS and other necessary libraries for it.
b. Understand & Modify codes. Under the instructions of TA, I had a rough idea of what those codes are for. Then I learned all relevant codes line by line, and modified them based on my experiment purpose. After this, I was able to run WProf embedded chrome to fetch websites, and analyze them into a .json file recording objects loading info and dependencies, as well as a summary file documenting important info as PLT and Critical Path.
c. Add codes and tool for DNS cache experiment. For local machine DNS cache control, I used the tool dnsmasq to control it. The hard problem is how to control the DNS cache on local DNS server. For this, my solution is to experiment with unpopular websites, which can ensure that there should be no DNS cache on both local machine and local DNS server the first time I load them. After loading them, wait about 10 mins and the cache on local DNS server should be flushed, and can be loaded repeatedly with no DNS cache.
d. Experiment 1. Experiment with 100 popular websites for 'cache/no cache', and 60 unpopular websites for 'dns cache/no dns cache', and evaluate the result in respect of the web load time for different parts of the loading.

e. Experiment 2. Experiment with 3 websites, one mainly consists of pictures, one of text, and one of Javascripts, with each website loaded 7 times for three conditions: with no http cache and DNS cache, with http cache and DNS cache, with no http cahce and no DNS cache. Then evaluate how http cache and DNS cache affect critical path.

3. Evaluation Methods

Evaluation method for Experiment 1:

Run 'sudo python3 yunke.py' and get the .json files and final summary files for 100 popular 100 websites and 60 unpopular websites (part of websites encountered the Chromium crash "Aw, snap!"). Then run 'sudo python3 visualize.py', which collects all important web loading information, computes statistics for them, compares them across four conditions ('cache/no cache', 'dns cache/no dns cache'), and output the scatter plot and CDF plot for comparison visualization.

The important information contains: load time, download time, computation time, block time, TTFB time, download html time, download css time, download img time, download js time.

Note that the result variance is large because of various unstable conditions, and the conclusions from the figure are more like reasonable guesses than assertions.

Evaluation method for Experiment 2:

To avoid the large variance, I repeatedly load pages and get the average to study the behavior of critical path under different conditions.

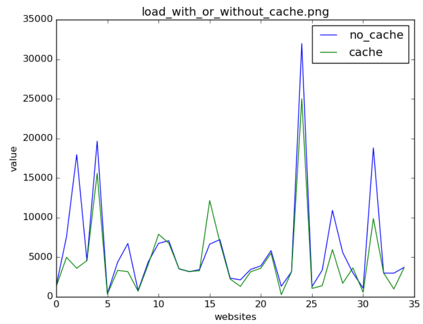
Run 'sudo python3 yunke2.py' and get the .json files and final summary files for 3 websites repeatedly 7 times across three conditions (with no http cache and DNS cache, with http cache and DNS cache, with no http cahce and no DNS cache). Then, use 'sudo python3 deep_analysis.py' to do further analysis about how local cache and DNS cache affect critical path. Generally, 'deep_analysis.py' collect data and analyze the the activities around critical path: how the average time of each type (image, text, application, evaluation, normal computation) of activities changes under different conditions; 'deep_analysis.py' also analyze how the percentage of each type in the critical path changes under different conditions.

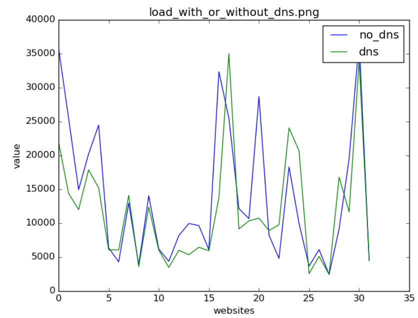
4. Results and Discussions

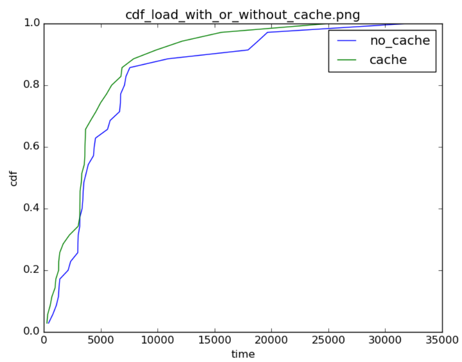
Experiment 1:

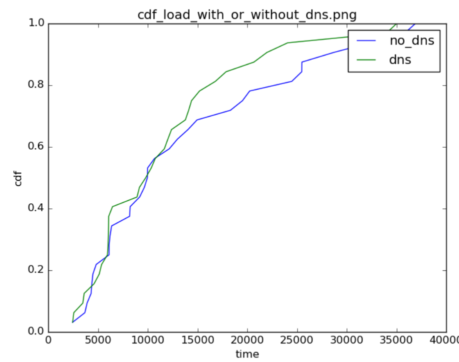
As I stated before, there's large variance because of various unstable environment. Also, I doubled the number of experiment websites (from 50 pop and 30 unpop, to 100 pop and 60 unpop). Below are some noticeable figures and discussions:

A. Total load time:





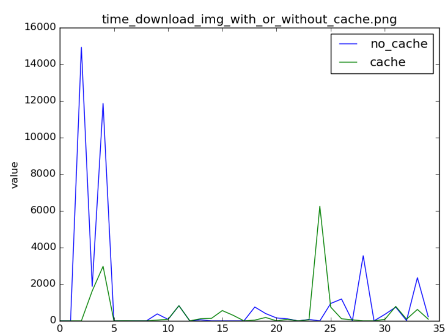


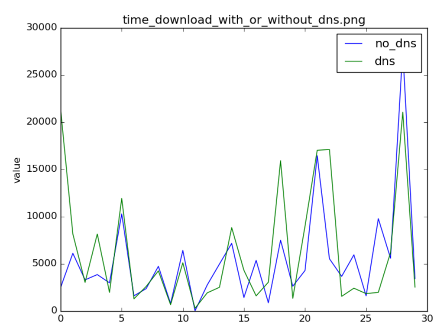


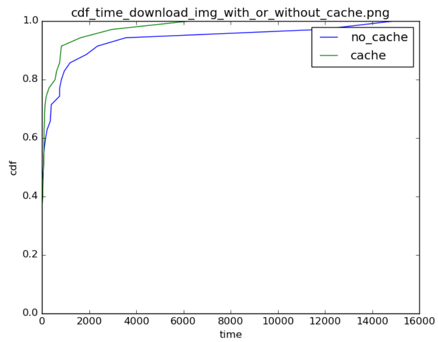
Local cache significantly reduces load time, because of the time saved when downloading (avg = 4624 with cache, avg = 6108 without cache)

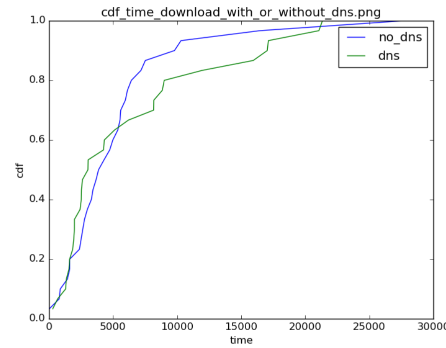
DNS cache very slightly reduces load time (avg = 11779 with DNS cache, avg = 13761 without DNS cache)

B. Download time:





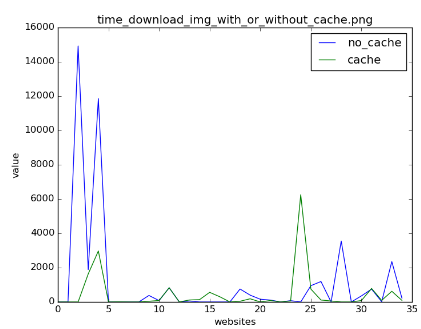


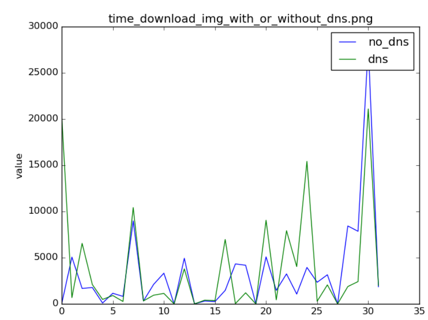


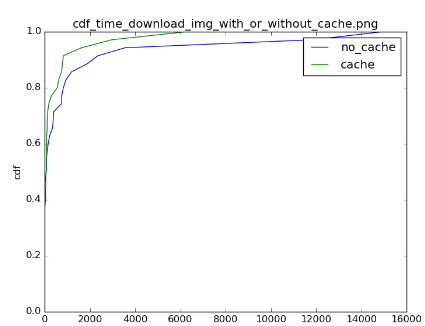
Local cache significant reduces download time, because that's what local cache is designed for. (avg = 1408 with cache, avg = 3017 without cache)

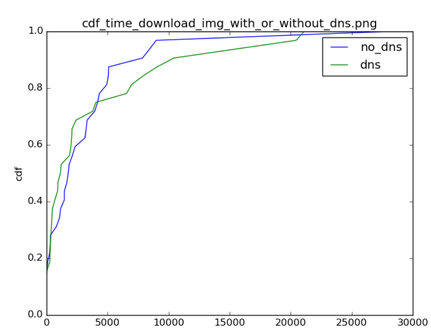
DNS cache has no obvious effect on download time. This is weird because DNS cache should have improved the download speed. However, I repeated this test several times, and also checked the code, and found no problem. I think the reason may be the algorithm of the analysis code which Javad provided: maybe the definition of the 'download time' is different from what I understand, or there's problem with the code, because there're very strange time computations in .json file (negative value, or very large value).

C. Download image time



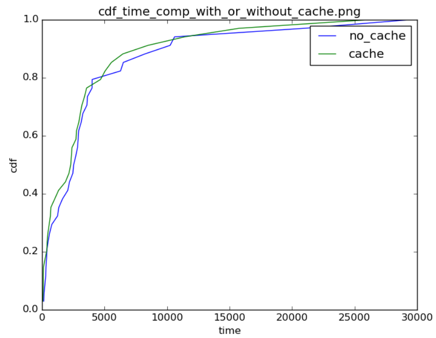


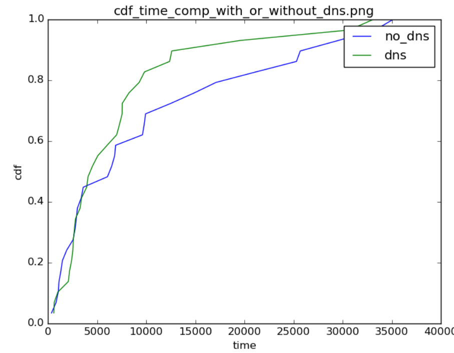




Similar conclusion with the above "download time".

D. Computation time





Local cache has a very slight improvement on computation. (avg = 3602 with cache, avg = 4191 without cache)

DNS cache has an improvement on computation. (avg = 7421 with DNS cache, avg = 10231 without DNS cache) This is hard to understand, because the browser should look up the resolution record on local machine first and this takes time to compute. Again, it may be the problem of the algorithm or definition of the analysis code.

Experiment 2:

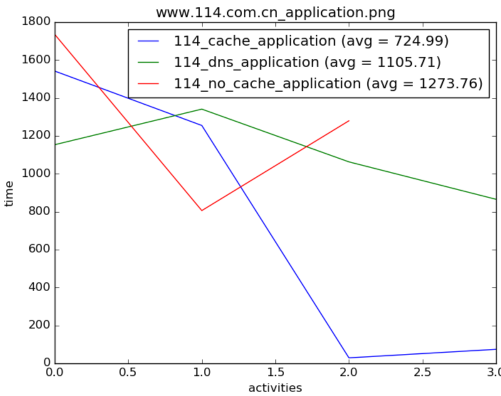
To have a deeper insight into how different cache condition affects critical path behavior, I experimented with 3 websites (www.114.com.cn consists of text, www.mi.com consists of js application, www.wmpic.me consists of images), and repeatedly several times to get the average to reduce variance.

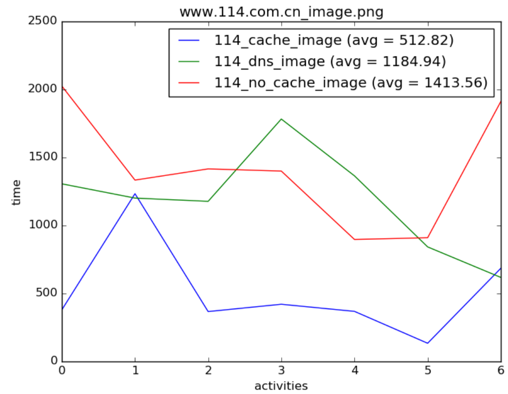
First, I found out the critical path related activities for each website, i.e. I union all critical path activities across all repetitions and under different cache conditions, and these activities should be close to the critical path.

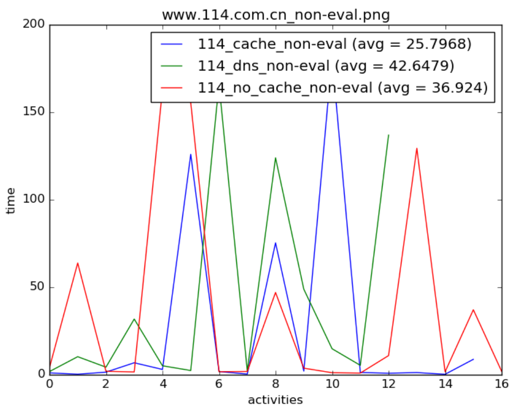
Then, for the four types, ‘image’, ’text’, ’application’, and ’non-evaluation computations’ (I didn’t consider the ‘evaluation computation’ because it frequently has time interval of negative value and extremely large value), I computed the average time for each ‘around critical path activity’ for each type, and scatter them.

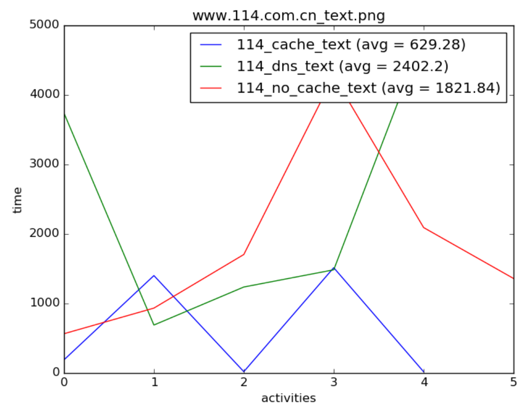
Finally, I calculated, for each website, under each cache condition, what’s the ratio of four types on critical path.

A. Average time

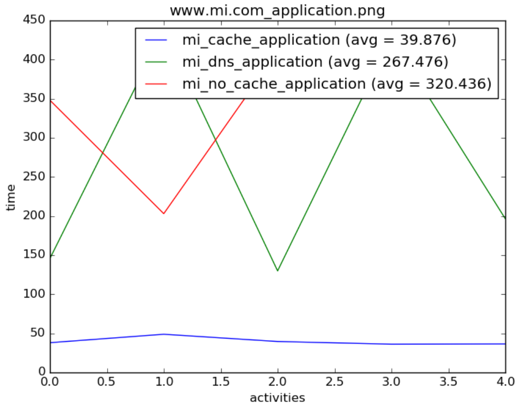


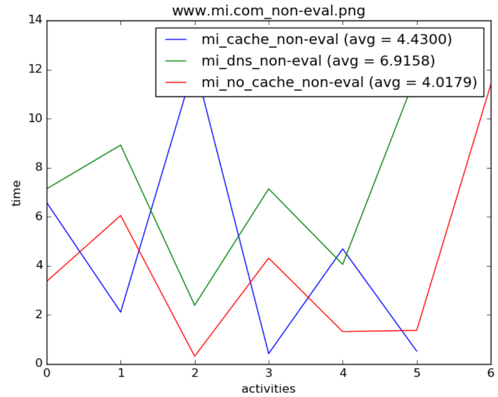


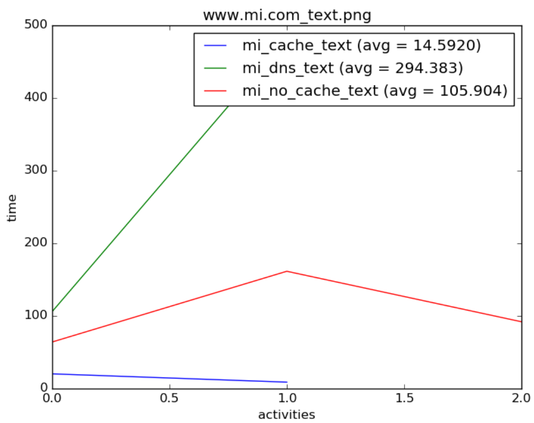




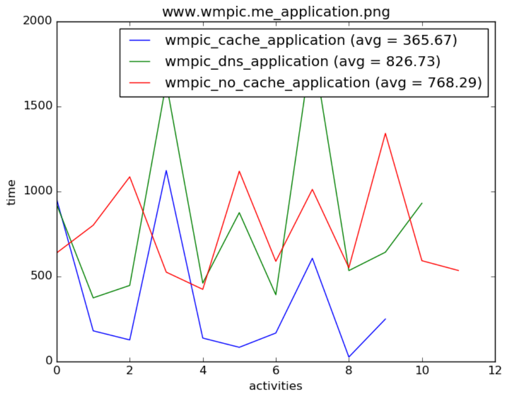
'114' mainly consists of text. As we can see, local cache largely reduces the text download time (including text/html, text/css, text/js, …) on critical path, and slightly reduces the non-evaluation time on critical path, while DNS cache increases text download time on critical path, and slightly reduces download time for other components.

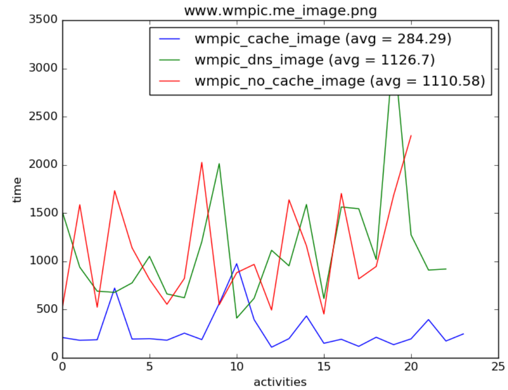


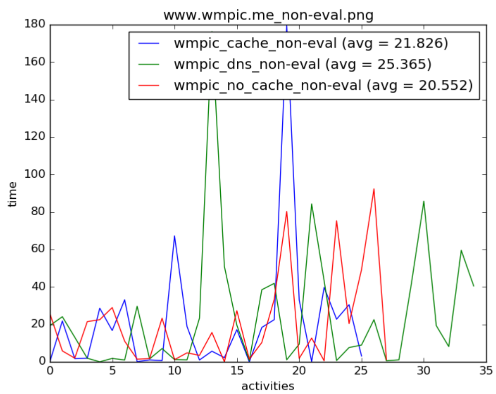


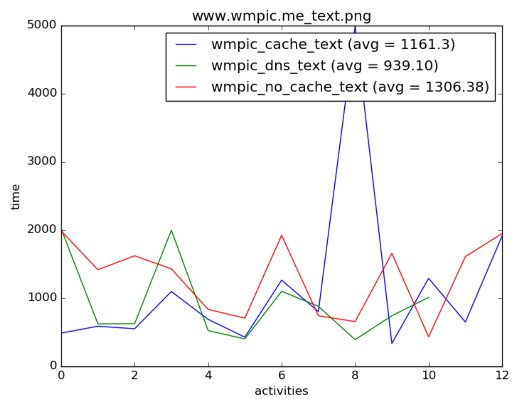


'mi' consists mainly of js application. As can be seen, the application download time has a large percentage. Again, local cache largely reduces application download time, while DNS cache only slightly reduces application time.



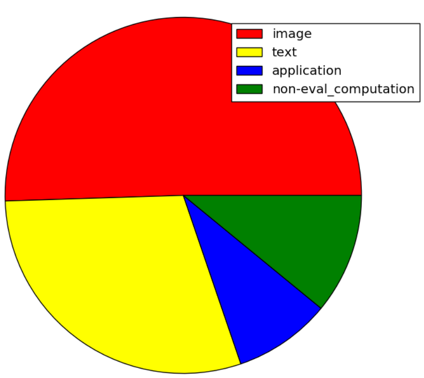




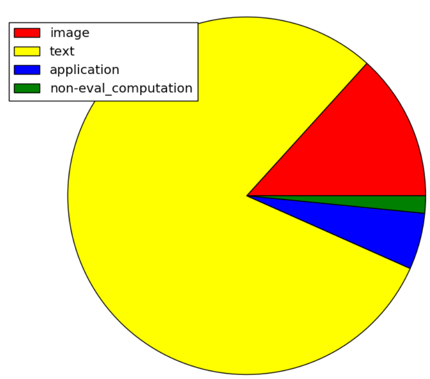


'wmpic' mainly consists of images. On the critical path of these websites, local cache significantly reduces image download time and application time, while have no obvious effect on text download time and non-evaluation computation time.

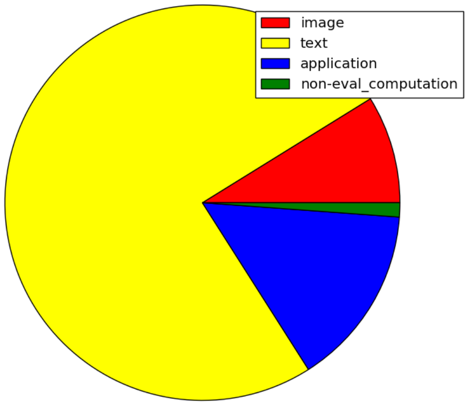
B. Type ratio



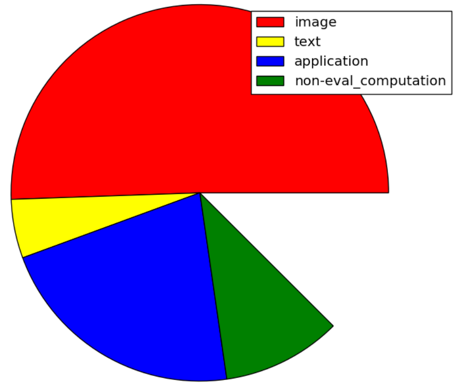
'114' with local cache



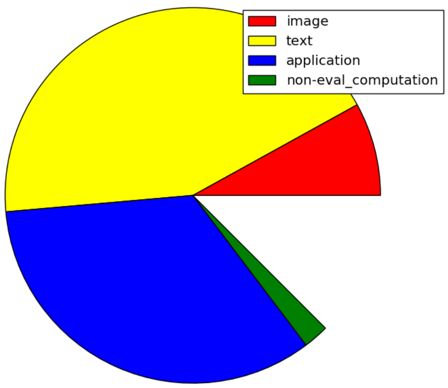
'114' with dns cache



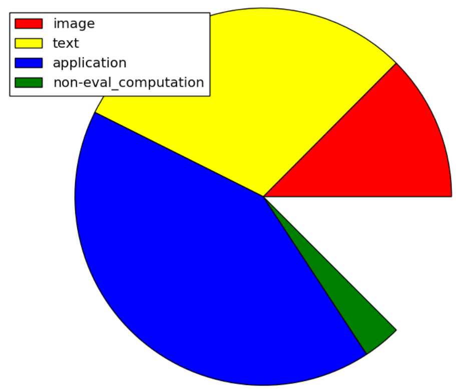
‘114’ without local cache



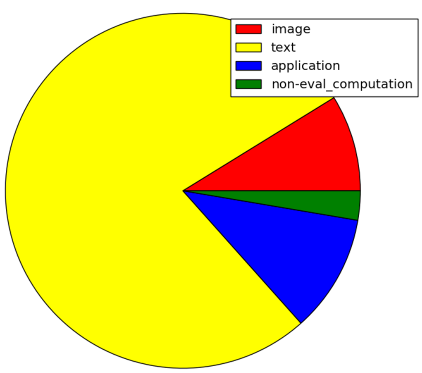
‘mi’ with local cache



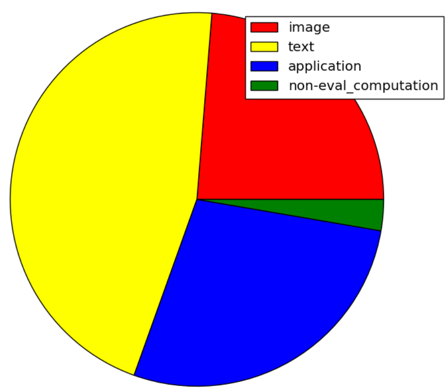
‘mi’ with dns cache



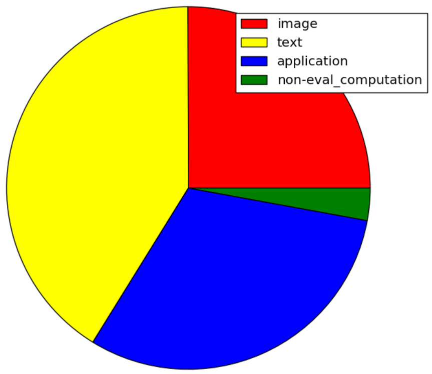
‘mi’ without local cache



‘wmpic’ with local cache



‘wmpic’ with dns cache



'wmpic' without local cache

From the pies above, we can conclude that:

a. For mainly text websites, local cache largely reduces the percentage of text, while reduces less for other components. DNS doesn't change the critical path type ratio much.
b. For mainly js application websites, local cache largely reduces the percentage of application and text, and image percentage becomes relatively high. DNS cache slightly reduces application percentage, and also reduces image percentage.
c. For mainly image websites, local cache largely reduces the percentages of image and application, while increases the percentage of text. DNS has very slight change to the ratio of each type.
d. As evaluation time occupies a large part of critical path, but there're too many invalid values and thus useless, so the behavior of computation is not studied, while 'non-evaluation computation' only accounts for a very small part, and hard to find patterns.

(relevant codes shared through google drive at the link: https://drive.google.com/a/stonybrook.edu/folderview?id=0B16l5ryEUECfWVpJYXFzU1pzeVE&usp=sharing if not accessible, please contact me at yunke.tian@stonybrook.edu. I'll send codes to you once I receive your quest)