Performance Evaluation of DNS Resolver

Mridul Ranjan, SBU ID - 111482003

This report explains the details of the three experiments conducted with DNS resolver built as part A of assignment 1 (which we will call mydig resolver), local DNS resolver and Google DNS resolver. 25 top sites were picked from alexa.com and average time of resolving DNS query for each domain was captured.

1 Set up

- Local DNS Ip - 207.244.82.25, 108.59.15.5
- Google Resolver Ip - 8.8.8.8
- No of websites - 25

2 Observations:

- Local DNS resolver is the fastest.
- Google DNS resolver is close to Local DNS resolver but doesn't beat it.
- DNS resolver from part A is slowest of the three.

3 Explanation

Local DNS resolver and Google DNS resolver does a lot of caching. So they need not to contact root server for every query and drill down the domain tree. This saves a lot of time to respond to any DNS query. However, Local DNS resolver has the benefit of being located in close proximity with client than Google DNS resolver. This is the reason why Local DNS resolver performs slightly better than Google's DNS resolver. However, mydig DNS resolver, built in part A of assignment, doesn't cache any query. So it takes most time. Also, it relies only on timeouts to try other server for query resolution. This results in a high response time.

4 Data Collected

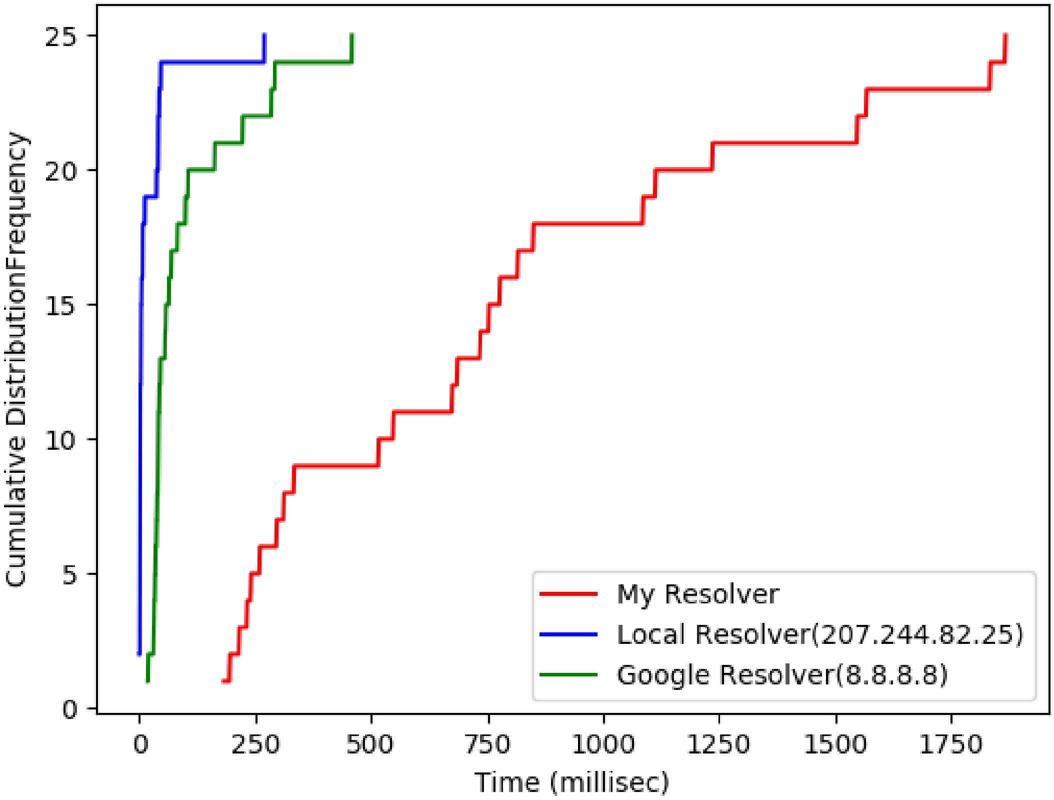
Websites Used for analysis - ['Google.com', 'Youtube.com', 'Facebook.com','Baidu.com','Wikipedia.org', 'Reddit.com','Yahoo.com', 'Google.co.in', 'Qq.com', 'Taobao.com','Amazon.com', 'Tmall.com','Twitter.com', 'Google.co.jp', 'Instagram.com', 'Live.com', 'Sohu.com', 'Vk.com', 'Sina.com.cn', 'Jd.com', 'Weibo.com', '360.cn', 'Google.de', 'Google.co.uk', 'Google.com.br']

My DNS Resolver - time in millisecs [549.5948791503906, 233.75766277313232, 818.1043386459351, 1570.5759048461914, 335.3635311126709, 196.9130277633667, 216.5031909942627, 1087.7935886383057, 687.812066078186, 313.409686088562, 242.07942485809326, 1237.6573085784912, 182.32088088989258, 778.4564733505249, 297.39606380462646, 259.96758937835693, 737.1018409729004, 1114.0063285827637, 1837.2947216033936, 517.2246932983398, 754.3838977813721, 1869.3859815597534, 1550.429344177246, 675.5406618118286, 851.6114473342896]

Local DNS Resolver - time in millisec [1.7240285873413086, 1.6727924346923828, 2.22780704498291, 2.3995161056518555, 7.576441764831543, 3.687763214111328, 0.3731727600097656, 1.6536235809326172, 3.8083553314208984, 12.46335506439209, 1.943063735961914, 43.15803050994873, 3.956460952758789, 269.94969844818115, 2.4312257766723633, 0.38700103759765625, 41.138243675231934, 1.8030881881713867, 46.20380401611328, 41.036128997802734, 7.397961616516113, 37.621188163757324, 2.392864227294922, 1.6207695007324219, 5.062413215637207]

Google DNS resolver - time in millise [57.431626319885254, 34.09576416015625,41.25092029571533, 40.77761173248291, 82.40177631378174, 42.04864501953125, 32.46879577636719, 55.64620494842529, 99.65531826019287, 43.18704605102539, 37.668561935424805, 105.51629066467285, 19.646644592285156, 64.49425220489502, 69.18301582336426, 35.446906089782715, 162.60502338409424, 18.393754959106445, 459.151554107666, 223.27568531036377, 292.4537658691406, 284.9555730819702, 38.67812156677246, 30.797314643859863, 45.38888931274414]

5 Cummulative Distribution Function Graph



The graph included in this section represents Cumulative Distribution Function(CDF) of the DNS resolution time for Experiment 1, Experiment 2, and Experiment 3. X-axis represents the time taken in milli secs for resolving DNS queries. Google DNS resolver is as good as Local DNS resolver will little latency over head because of being located remotely. Almost all the queries are resolved under 300 milli secs by Local and Google DNS resolver. Comparing it to the mydig DNS resolver, this difference is quite huge. This is because of no caching being done in mydig resolver and also reliability on timeouts which are added to query response time for bad servers.
Note : I/O was taking a substantial amount of time after resolving DNS query, so these data were captured by disabling all I/O operations.