

An Analytic Research on HTTP ,Web Servers, Web-Services & Efficient DNS Querying

Shaik Khaja Hafeezuddin¹, P.V.S.Srinivas²

¹UG Scholar, Department of Computer Science & Engineering, TKR College of Engineering & Technology, Hyderabad, India

²Professor,Head of the Department, Department of Computer Science & Engineering, TKR College of Engineering & Technology, Hyderabad, India

Abstract: *This electronic document is intended to give an overview on HTTPS, web-services, web-protocols, and their interaction with DNS Servers for efficient query processing. Efficiency is achieved by directly sending the query to top level Domain name servers. These top level domain name servers database and cache memory containing the address of the hosting systems, replies authoritatively to the user's query with the web-hosting address. Thus query sent by the user is processed efficiently in time by the top level domain name servers. The numbers of top level domain name servers are to be increased accordingly to handle the requests made by the users.*

Keywords: HTTPS, Web-services, Web-servers, Web-protocols, DNS, DNS Querying

1. Introduction

Web services provide a standard, reliable and efficient means of interoperating between different software applications, running on a range of different platforms and frameworks. This document is intended to provide a basic definition of a Web service, and define its place within a larger Web services framework to guide the technical community. The relationships among HTTP, Web Server and Web Services are a complicated set of functionalities where exchange of information takes place very rapidly and with zero scope for redundancy and information loss. Each of these segments are very clearly explained and documented such that they are understandable by the beginners too. Each component have many important roles and thousands of functions which users can access and utilize on the Internet. HTTP allows users to interact with Web Servers and access information through the Internet. Web servers are the source to serve data and files to users who request them. Web Services allow cross-system, cross-language, cross-language communication among various kinds of machines and thus enabling wide range of inter-business transaction. Each technology works on its own set of principles, have own set of requirements, and also performs many useful functions as per user requirement; it is the combination of these technologies that has created the dynamic functionalities of the Web which are available today. This research paper will explain in depth about the Interrelationship between HTTP, Web Servers and Web Services technologies that have facilitated the functionalities and convenience of the Web.

2. HTTP

HTTP, is the standard protocol which is currently used to access the Internet. According to the World Wide Web Consortium, "HTTPS is an application-level protocol for distributed, collaborative, information systems or machines." [1]. It is a very simple protocol that allows raw data to be transferred over the Internet in the form of packets. From this simple data transfer protocol, users of the Internet can easily perform functions and give commands to

the Web Servers through a GUI.As Web page viewed through a browser and the users need not worry about the details and the process of how the command is going to be transferred or interpreted by the computers involved within the network. HTTP is enhanced with a facility of exchanging the information between the user's computer and the Web Server rapidly and efficiently.

As HTTP allowed the World Wide Web and the Internet to become such a global need, the World Wide Web and the Internet also helped HTTP to become the global standard of data transfer protocol over the web platform. The World Wide Web global adopted HTTP as the basic data transfer protocol from the year 1990, allowing HTTP to develop into a global standard of data transfer. However, just because HTTP is the current global standard it does not mean that it is flawless or do not have any short comes. An example of HTTP's drawbacks would be that HTTP is a stateless protocol, which means HTTP treats each command independently. In another words, HTTP does not have any storage or cache memory. HTTP does not remember past commands and will forget the current command as soon as it is executed. As such, it is very hard to make HTTP to be more interactive and dynamic without additional technologies such as JavaScript, cookies, PHP and other programming scripts. To put it simply, with pure HTTP user will not be able to customize their online shopping cart.

Many efforts have been made to improve HTTP. From the first HTTP 0.9, through HTTP 1.0, to the latest version HTTP 1.1, numerous and many countable improvements have been made. Some of them are:

- Faster response.
- Greater bandwidth savings, by adding cache support.
- Faster response for dynamically-generated web pages, which supports chunked encoding, and allows a response to be sent before its total length is known.
- Efficient use of IP addresses available.

3. Web Servers

A server is a system or a machine or a device on a network that manages network resources accordingly.[2]. There are many different kinds of servers which are categorized based on the requirements, dataset, and usage. They are the dial-up servers that serves as a gateway for the user to access the Internet; a printer server, and a Web Servers or information storing servers that stores web pages and other data and information that are provided to users upon request or when the query is passed to the server to serve the information.

A Web Server is the central system of Web Site i.e a web server is a place where the pages related to the website are hosted. These web pages are served to the user when the user requests the data through the browser. It is the Web Server that hosts both the components of a Web page such as the actual Web page HTML files, JSP files, JQUERY files, CSS files and templates and all other essential technologies that make a Web site function in a way that it has to be. All the Web servers may function similarly, the way of setting up and the way a server can be varied accordingly.

There are two mostly used ways in setting up Web Server: One is P2P type of set up, and other is Client-Server type of setup. P2P (Peer-to-Peer), follows a direct connection of individual computers to one another where each computer can specify what data it is willing to share with the other computer on a network. [3].This kind of web server set up is very easy and cheap. Further, the speed of file transfer in a P2P type network is not constrained by the capability of any single server. Here each computer in the network is capable of becoming a server, such that it is able to share and transfer a file from multiple servers at the same time, this in turn increases the file transfer rate between the two nodes. However, since each computer in the network is a server, each computer on a particular network needs to be set up manually to achieve the intended functionality. The responsibility of managing the server or system lies with the owner of a computer that is connected to the network. But, the management of a P2P network is very much difficult. Due to the decentralized management of the network, servers with a P2P connection are prone to virus and worm attacks, which if affected may collapse the entire network over any topology.

In a Client-Server network, on the other hand, it is a highly centralized network system with one main central computer as the server unlike P2P type. This type of set up is easy to manage and is very secure when compared to others. Yet, maintaining a centralized network requires much amount of resources ranging from huge manpower to high hardware. This results in increase in the cost of a client-server network. Another drawback of a client-server set up is that the speed of file transfer between the client and the server slows down when the number of clients accessing the server at a time increases. Exception is that, it is very easy to manage and is very much secure, client-server network is still the dominant set up of Web Servers. Apache, which is a free server technology, is currently one of the most popular server technologies because of its ease of usage and flexibility till date.

The first version of Apache is based on the NCSA HTTPd Web Server, which was developed in 1995 by a “loosely-knit group of programmers (20 in number).” [4].Apache server provides full source code and an unrestrictive license to users. Apache users can easily change, modify, or adapt, the software in accordance with the needs of their particular organization or as per their requirements. There is an option of Additionally many modules, either written by the user or downloaded free of charge from the vast Apache module library online database, could easily be added to accommodate any specific needs of the users. Apache is also capable of performing many functions such as DBM database unlimited flexible URL rewriting and aliasing, authentication, multiple Directory Index directives, content negotiation and virtual hosts.

4. Web Services

Web Service is a very powerful tool that has enhanced the efficiency in communication among fields and domains. According to the World Wide Web Consortium, a Web Service is a intelligent software system meant to support inter-operable machine-to-machine interaction over and within a network. Web Services is a software system that allows systems & machines (including servers) to communicate with each other regardless of each individual machine’s operating systems and programming languages. According to Symon's Extensible Markup Language (XML) page there is a formula that neatly defines the major components of Web Services: “Web services = XML + SOAP + WSDL + UDDI” [6].

Extensible Markup Language (XML) is the universal markup language that all machines are capable of understanding. The process of inter-machine communication via Web services, XML is used to tag the data involved in the communication between two communicating nodes. Web Services Description Language, is being used for describing the services available. The Universal Description - Discovery & Integration list out the services available from a particular machine. Simple Object Access Protocol is used to transfer data for each exchange of information between machines and servers, which typically involve HTTP in conjunction with XML serialization other Web-related standards. Web Services are completely independent of operating system, programming language.” As a result, through the Web Services, there is a facility of Java based programs will be able to communicate to servers that are running C++ based programs and a Windows machine will be able to communicate with a LINUX machine or server. Which means web services and completely platform independent.[8].

Though web and a web service serve a similar function, Services do have some most significant differences. One of the most prominent differences between Web services and the Web is that instead of a user interface, Web Services functions through/via application interfaces. In other words, it can be stated as the machines communicate with each other from application to application between two active nodes. Here communication takes place between the applications of one machine to the other machine. Such

exchanges limit possible user errors and thus increase the efficiency of the exchange.

5. How HTTP, Web Servers and Web Services Interact Together

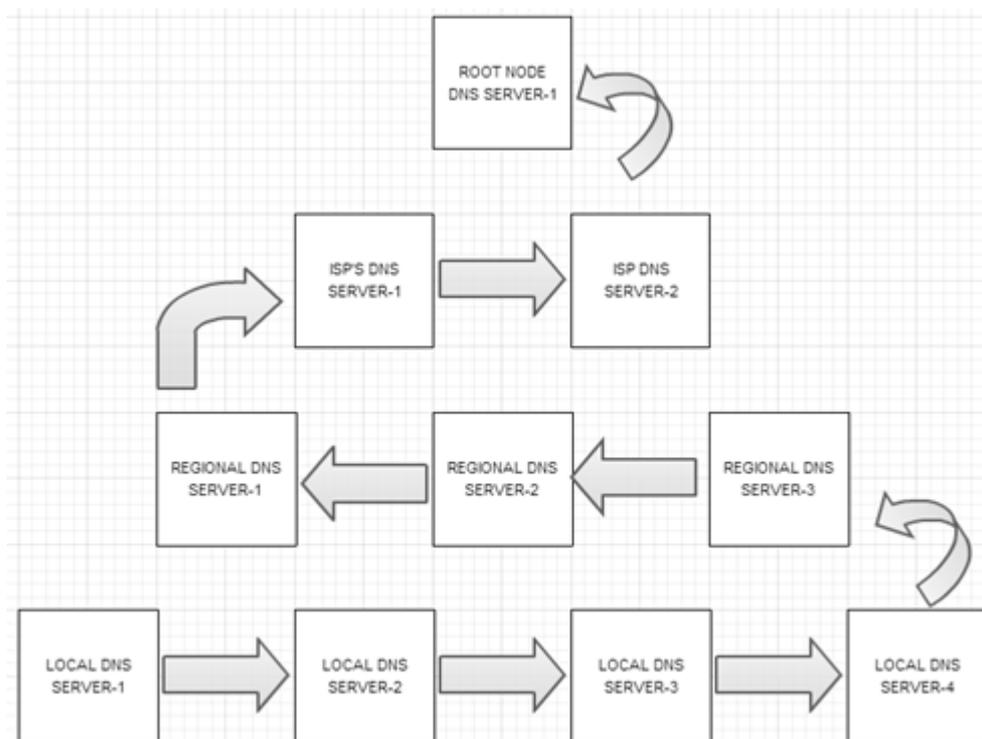
The interaction between HTTP, Web Servers and Web Services is very simple and can be documented in a lucid way as follows: HTTP is a simple protocol through which browsers use to communicate with Web Servers. Web Servers, on the other hand are responsible for fulfilling the user's request's and store the information users provide. Web Services allow different Web Servers to communicate and interact with another in order to process the request and/or commands given by user through browser or any GUI.

The interconnectivity among the three technologies works can be explained by using an example where a user is trying to buy a plane ticket online. The user will be accessing a travel agency web site to query for the availability of seats on the required date and time. Here HTTP acts as the language through which the users will be able to communicate with the Web server that actually can access the information of flight date, time, seats, price from the airlines database. According to the values given by the user to the Web page and transmitted to the Web server through

HTTP protocol, the Web server performs the command of search by sending out the commands of this query to each individual airline's schedule databases using an application to application interface (Web services). Web services translate the markup language the Web server uses into the universally understood XML that is given as input to the databases of all the available active airlines. When the XML is received by the airline databases, Web services translates the xml into the programming language that each one of airline database is using so that the database would be able to understand the command the Web server sent out. After the execution of the query, the result would be transmitted back to the Web server through Web services. Then the Web server would be able relay these search results to the user through HTTP which would present the information to the user through an HTML file that could be interpreted by any of the browsers.

In total, the functions of querying requires all three technologies, HTTP, Web server and Web services, have to work together for a success full outcome. Without any one of these technologies, the query would fail or may have very limited scope.

5.1 Request processing by DNS servers Traditional

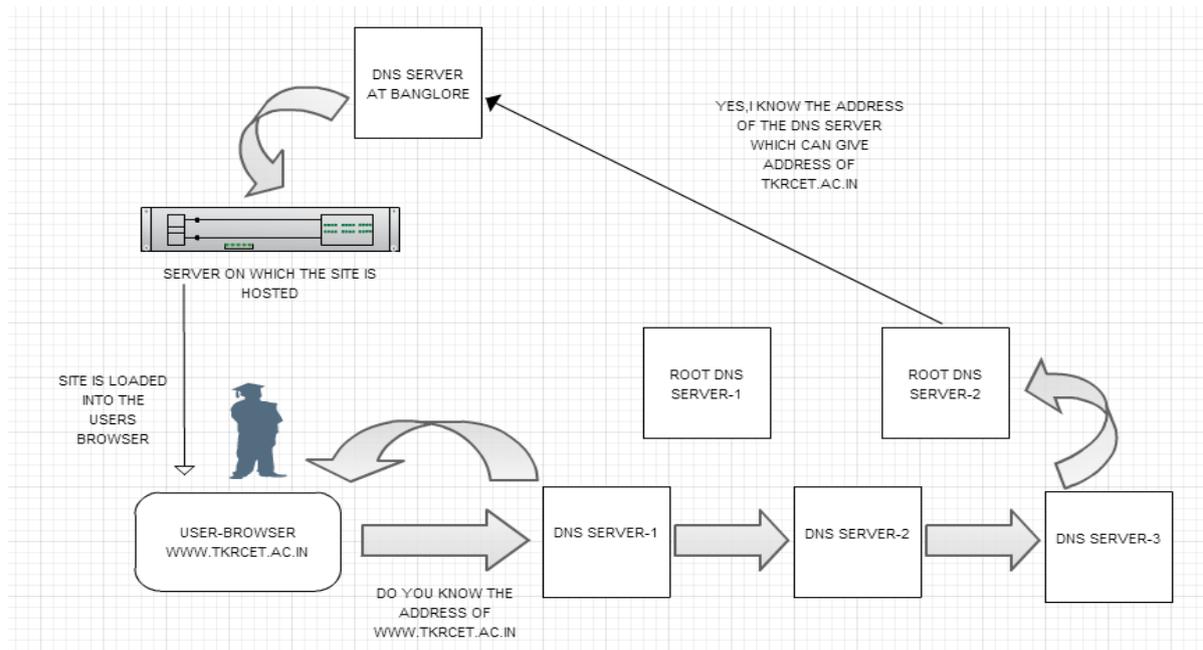


When the DNS servers receives a query, it first checks and see if it can answer the query based on resource information and available records that contained in a local zone of the server. If the queried name matches a corresponding resource record in local zone information, the server answers to the user request using this information to resolve the queried name. If no information exists for the queried name by the user in a particular zone, the server then checks to see if it can resolve the name using the local cached information

from previous queries which it stores when it gets the answers from the other servers for a particular query. If any match is found, the server answers with this information. Again, if no match is found the process continues till it reaches the top level domain name servers where every address can be fully resolved as these top level (total 13 in number across the world) DNS servers holds all the data of addresses zone wise. By default, the DNS Client service asks the server to use a process of recursion to fully resolve

names on behalf of the client before returning an answer. In many cases, the DNS server network is configured, to

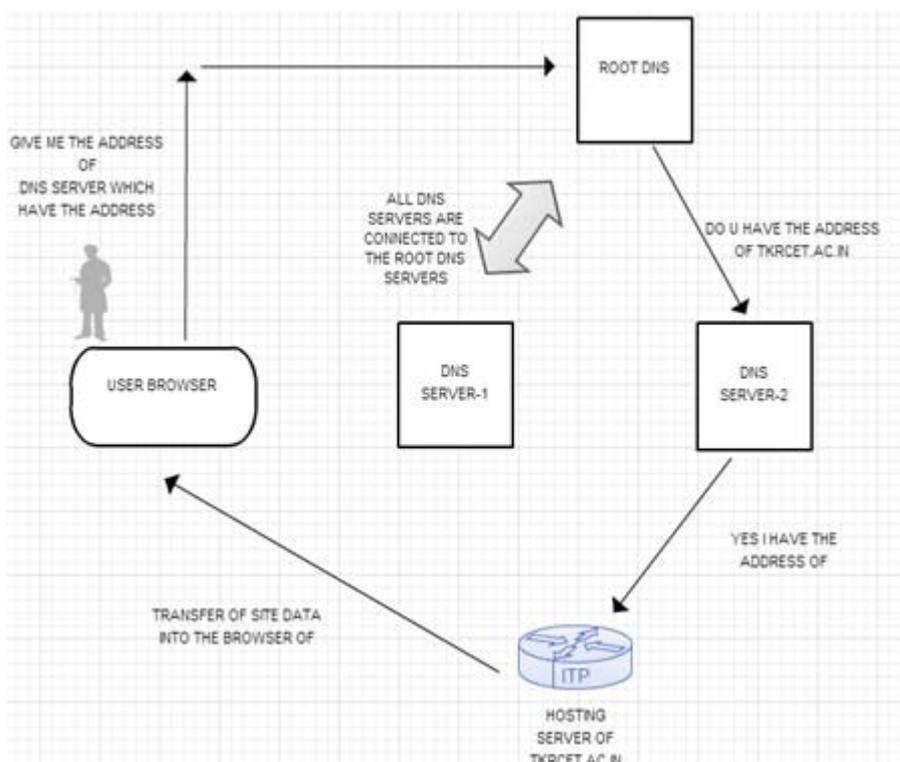
support the recursion process as shown in the following figure



5.2 Proposed DNS Querying Model

When the local DNS server receives a query, it first redirects the query to the top level domain name servers where the address is checked for the relevant location of the local

server address on which the site or webpage is hosted. After locating the hosting server the DNS server answer the query authoritatively based on resource record information. The user is redirected to the address of the local server which serves the user request.



6. The Role of Humans

Although the main purposes of Web services is to automate processes that might otherwise be performed manually, humans still plays an important role in their architecture designing, maintaining and use, notably in two ways:

1. Humans have to agree on the semantics and the service description. Since a humans (or let's call an organization) is the legal owner of any Web service, people must agree on the semantics, usage agreement and the service description that will govern the interaction with the web-server through web services. Often this

agreement is accomplished by the provider entity and offering both the semantics and the service description as accept it or reject it contracts that the requester entity must accept as mentioned by the legal entity, unmodified as conditions of use. In order to use the services provided by the owner the user must agree to the terms and conditions of the service provider. However, nothing in this architecture prevents them from reaching agreement by other sort of means.

2. User creates the request and provider agents either directly or indirectly. Humans, the users must ensure that these agents implement the terms of the agreed-upon service description agreements and semantics. Regardless of the approach or form used, from information point of view both the semantics and the service description must be somehow be input and both the service providing agent and the provider agent before the two agents can interact.

7. Conclusion

The functionalities that HTTP, Web Servers and Web Services provide now in the current scenario dramatically changed the way individuals, companies, and the people conduct business online. While it will be suitable to state that each technology was created for one specific purpose, the combination of these technologies that has greatly enhanced the transfer of information online. The example of users purchasing tickets online shows how critically important is a role of each technology, all these plays a very prominent role in one of the most common tasks users can accomplish on the Internet today. Without any one of these technologies, e-commerce industry all over the world would not have boomed as much as it is now and the convenience users may not have increased to this extent without these technologies.

References

- [1] <http://www.w3.org/Protocols/rfc2616/rfc2616-sec1.html#sec1>
- [2] (<http://www.jmarshall.com/easy/http/#http1.1>)
- [3] ([http://www.hchs.hc.edu.tw/cc\[4\]http://www.webopedia.com](http://www.hchs.hc.edu.tw/cc[4]http://www.webopedia.com))
- [4] <http://www.w3.org>
- [5] <http://home.earthlink.net>
- [6] Marshall, J. (1997). *HTTP made really easy: a practical guide to writing clients and servers*.
- [7] Ryman, A. (2003). *Understanding Web Services*.
- [8] Webmonkey (1999). *Introduction to Apache*.
- [9] http://www.webopedia.com/TERM/A/Apache_Web_server.html
- [10] <http://www.webopedia.com/TERM/s/server.html>
- [11] Webopedia (2003). *Web Services*