International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 12
April 2018

DNS for Query: efficiency and security

P.Kalpana , B.Sravya , M.Supraja , Ch.Pruthvi
Assistant Professor in Department of Information Technology .in Teegala Krisha Reddy Engineering college.Telangana
gonakalpanarao@gmail.com
UG Scholar in Department of Information Technology .in Teegala Krisha Reddy Engineering college.Telangana
UG Scholar in Department of Information Technology .in Teegala Krisha Reddy Engineering college.Telangana
UG Scholar in Department of Information Technology .in Teegala Krisha Reddy Engineering college.Telangana

AbstractAbstract*—Web search engines are composed by thousands of query processing nodes, i.e., servers dedicated to process user queries. Such many servers consume a significant amount of energy, mostly accountable to their DNSs, but they are necessary to ensure low latencies, since users expect sub-second response times (e.g., 500 ms). However, users can hardly notice response times that are faster than their expectations. Hence, we propose the DNS Algorithm to select the most appropriate DNS frequency to process a query on a per-core basis.*

Keywords—DNS, Web search engines.

I.INTRODUCTION

Web search engines continuously crawl and index an immense number of Web pages to return fresh and relevant results to the users' queries. Users' queries are processed by query processing nodes, i.e., physical servers dedicated to this task. Web search engines are typically composed by thousands of these nodes, hosted in large datacenters which also include infrastructures for telecommunication, thermal cooling, fire suppression, power supply, etc.

Web search engine datacenters, the query processing activity, and the query efficiency predictors. formulates the problem of minimizing the energy consumption of a query processing node while maximizing the number of queries which meet their deadlines. Our proposed solution to the problem, describes our DNS predictors, and the DNS optimization algorithm.

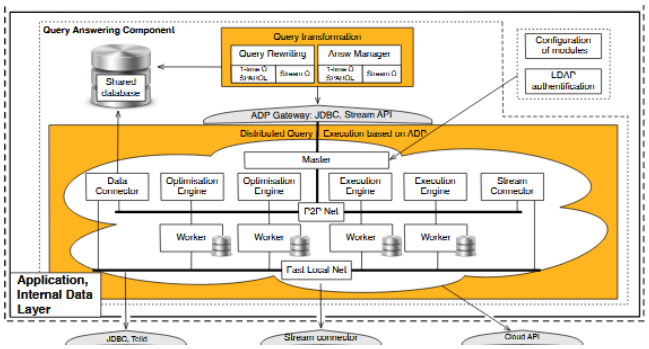


Figure: Query Processing .

II.MOTIVATION

In the past, a large part of a datacenter energy consumption was accounted to inefficiencies in its cooling and power supply systems. However, report that modern datacenters have largely reduced the energy wastage of those infrastructures, leaving little room for further improvement. On the contrary, opportunities exist to reduce the energy consumption of the servers hosted in a datacenter. In particular, our work focuses on the DNS power management of query processing nodes, since the DNSs dominate the energy consumption of physical servers dedicated to search tasks. In fact, DNSs can use up to 66% of the whole energy consumed by a query processing node at peak utilization.

Query processing and dynamic pruning Web search engines continuously crawl a large amount of Web pages. This collection of documents is then indexed to produce an inverted index . The inverted index is a data structure that maps each term in the document collection to a posting list, i.e., a list of postings which indicates the occurrence of a term in a document. A posting contains at least the identifier (i.e., a natural number) of the document where the term appears and its term frequency, i.e., the number of occurrences of the term in that particular document. The inverted index is usually compressed and kept in main memory to increase the performance of the search engine .

The Domain Name System specifies a set of various types of resource records (RRs), which are the basic information elements of the domain name system. Each record has a type (name and number), an expiration time (time to live), a class, and type-specific data. Resource records of the same type are described as a *resource record set* (RRset). The order of resource records in a set, which is returned by a resolver to an application, is undefined, but often servers implement round-robin ordering to achieve load balancing. The Domain Name System Security Extensions (DNSSEC), however, work on the complete set of resource record in canonical order.

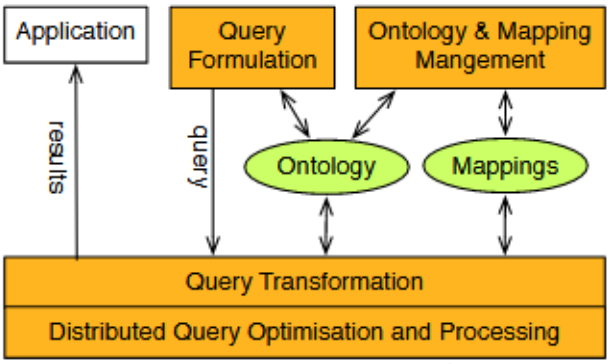
The Domain Name System delegates the responsibility of assigning domain names and mapping those names to Internet resources by designating authoritative name servers for each domain. Network administrators may delegate authority over sub-domains of their allocated name space to other name servers. This mechanism provides distributed and fault tolerant service and was designed to avoid a single large central database.

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 12
April 2018

The Domain Name System also specifies the technical functionality of the database service that is at its core. It defines the DNS protocol, a detailed specification of the data structures and data communication exchanges used in the DNS, as part of the Internet Protocol Suite. Historically, other directory services preceding DNS were not scalable to large or global directories as they were originally based on text files, prominently the hosts file.

III.PROPOSAL OVER VIEW

Query efficiency predictors (QEPs) are techniques that estimate the execution time of a query before it is actually processed. Knowing in advance the execution time of queries permits to improve the performance of a search engine. Most QEPs exploit the characteristics of the query and the inverted index to pre-compute features to be exploited to estimate the query processing times.owner. If someone try to copy the URL, the data get leaked in someone's laptop. Now the details about the unauthorized person will be tracked. This tracked information sent as a mobile intimation to the data owner. The mobile intimation will hold informations like IP address, MAC address and GPS location. Now we describe the results of the experiments conducted processing the realistic query workload. In this subsection we will not investigate research question based as for these experiments we use only the demand orinted retrieval strategy, which provided the best results in the following figure.



Firstly, we will analyze the performance of the three baselines. Then, we will discuss the results obtained by PROCESS in the time conservative configuration.

DNS responses traditionally do not have a cryptographic signature, leading to many attack possibilities; the Domain Name System Security Extensions (DNSSEC) modify DNS to add support for cryptographically signed responses. DNSCurve has been proposed as an alternative to DNSSEC. Other extensions, such as TSIG, add support for cryptographic authentication between trusted peers and are commonly used to authorize zone transfer or dynamic update operations.

In a *recursive query*, a DNS resolver queries a single DNS server, which may in turn query other DNS servers on behalf of the requester. For example, a simple stub resolver running on a home router typically makes a recursive query to the DNS server run by the user's ISP. A recursive query is one for which the DNS server answers the query completely by querying other name servers as needed. In typical operation, a client issues a recursive query to a caching recursive DNS server, which subsequently issues non-recursive queries to determine the answer and send a single answer back to the client. The resolver, or another DNS server acting recursively on behalf of the resolver, negotiates use of recursive service using bits in the query headers. DNS servers are not required to support recursive queries.

The *iterative query* procedure is a process in which a DNS resolver queries a chain of one or more DNS servers. Each server refers the client to the next server in the chain, until the current server can fully resolve the request. For example, a possible resolution of www.example.com would query a global root server, then a "com" server, and finally an "example.com" server.

Some domain names may be used to achieve spoofing effects. For example, paypal.com and paypa1.com are different names, yet users may be unable to distinguish them in a graphical user interface depending on the user's chosen typeface. In many fonts the letter *l* and the numeral *1* look very similar or even identical. This problem is acute in systems that support internationalized domain names, as many character codes in ISO 10646 may appear identical on typical computer screens. This vulnerability is occasionally exploited in phishing.

Domain name resolvers determine the domain name servers responsible for the domain name in question by a sequence of queries starting with the right-most (top-level) domain label.

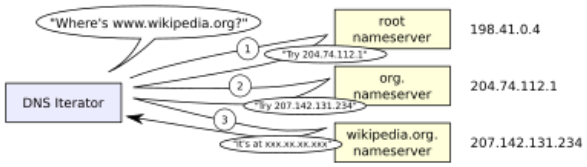


Figure: address resolver

For proper operation of its domain name resolver, a network host is configured with an initial cache (*hints*) of the known addresses of the root name servers. The hints are updated periodically by an administrator by retrieving a dataset from a reliable source.

Name servers in delegations are identified by name, rather than by IP address. This means that a resolving name server must issue another DNS request to find out the IP address of the server to which it has been referred. If the name given in the delegation is a subdomain of the domain for which the delegation is being provided, there is a circular dependency.

In this case, the name server providing the delegation must also provide one or more IP addresses for the authoritative name server mentioned in the delegation. This information is called *glue*. The delegating name server provides this glue in the form of records in the *additional section* of the DNS response, and provides the delegation in the *authority section* of the response. A glue record is a combination of the name server and IP address.

Assuming the resolver has no cached records to accelerate the process, the resolution process starts with a query to one of the root servers. In typical operation, the root servers do not answer directly, but respond with a referral to more authoritative servers, e.g., a query for "www.wikipedia.org" is referred to the *org* servers. The resolver now queries the servers referred to, and iteratively repeats this process until it receives an authoritative answer. The diagram illustrates this process for the host that is named by the fully qualified domain name "www.wikipedia.org".

This mechanism would place a large traffic burden on the root servers, if every resolution on the Internet required starting at the root. In practice caching is used in DNS servers to off-load the root servers, and as a result, root name servers actually are involved in only a relatively small fraction of all requests.

Techniques such as forward-confirmed reverse DNS can also be used to help validate DNS results.

IV. CONCLUSIONS

We have shown efficient execution of queries on big data is an open research problem and initial results achieved by research prototypes such as DNS are encouraging. In the Optique project we will push the barrier and provide massively parallel and elastic solutions for query optimisation and execution over Big Data integration. Our solutions based on ground breaking research will be deployed and evaluated in our use cases.This will provide valuable insights for the application of semantic technologies to data integration problems in industry. DNS optimization with baseline.Futher with consideration of query processing using DNS time can be done with deadlock prevention during resource sharing.

REFERENCES

[1]. Abouzeid, A., Bajda-Pawlikowski, K., Abadi, D.J., Rasin, A., Silberschatz, A.: HadoopDB: An architectural hybrid of MapReduce and DBMS technologies for analytical workloads.PVLDB 2(1), 922–933 (2009)

[2]. Calvanese, D., Giacomo, G.D., Lembo, D., Lenzerini, M., Poggi, A., Rodriguez-Muro, M.,Rosati, R., Ruzzi, M., Savo, D.F.: The MASTRO system for ontology-based data access.Semantic Web 2(1), 43–53 (2011)

[3]. Crompton, J.: Keynote talk at the W3C Workshop on Semantic Web in Oil & Gas Industry: Houston, TX, USA, 9–10 December (2008), available from

[4]. Dean, J., Ghemawat, S.: MapReduce: simplified data processing on large clusters. Communications of the ACM 51(1), 107–113 (2008),

[5]. O., Rodriguez Muro, M., Rosati, R., Schlatte, R., Schmidt,

M., Soylu, A., Waaler, A.: Scalable End-user Access to Big Data. In: Rajendra Akerkar: Big Data Computing. Florida : Chapman and Hall/CRC. To appear. (2013) 7. Greenplum:

”greenplum, http://www.greenplum.com/” (2011),

[6]. processing flows on the cloud. In: Proc. of SIGMOD. pp. 289–300 (2011) Rodriguez-Muro, M., Calvanese, D.: High performance query answering over dl-lite ontolo-

gies. In: KR (2012) Thusoo, A., Sarma, J.S., Jain, N., Shao, Z., Chakka, P., Zhang, N., Anthony, S., Liu, H.,

Murthy, R.: Hive - a petabyte scale data warehouse using Hadoop. pp. 996–1005 (2010)14. Tsangaris, M.M., Kakaletris, G., Kllapi, H., Papanikos, G., Pentaris, F., Polydoras, P.,

[7]. Sitaridi, E., Stoumpos, V., Ioannidis, Y.E.: Dataflow processing and optimization on grid

and cloud infrastructures. IEEE Data Eng. Bull. 32(1), 67–74 (2009)