Load Distribution among Replicated Web Servers: A QoS-based Approach

Marco Conti
Consiglio Nazionale delle Ricerche
Via Santa Maria 36
56126 Pisa, Italy
Tel. +39 050 593258
m.conti@cnuce.cnr.it

Enrico Gregori
Consiglio Nazionale delle Ricerche
Via Santa Maria 36
56126 Pisa, Italy
Tel. +39 050 593250
e.gregori@cnuce.cnr.it

Fabio Panzieri
Università di Bologna
Mura Anteo Zamboni 7
40127 Bologna, Italy
Tel. +39 051 354508
panzieri@cs.unibo.it

ABSTRACT

A dominant factor for the success of an Internet based Web service is the Quality of Service (QoS) perceived by its users. The principal QoS attributes these users perceive include those related to the service "responsiveness", i.e. the service availability and timeliness. In this paper, we argue that QoS can be provided by distributing the processing load among replicated Web servers, and that these servers can be geographically distributed across the Internet. In this context, we discuss strategies for load distribution, and summarize a number of alternative architectures that can implement those strategies. The principal figure of merit we use in order to assess the effectiveness of the load distribution strategies we discuss is the response time experienced by the users.

Keywords

Web server, Load Distribution, QoS.

1. INTRODUCTION

A practical approach to the provision of responsive Web services has been proposed in [8]. This approach is based on introducing redundancy, in the implementation of a Web service, by replicating Web servers (WSs) at distinct sites, and ensuring that each client (or browser) gets bound to the "most convenient" WS replica. In this context, the phrase "most convenient" is used to characterize that particular replica, among those that implement the same service, that can provide a client with the shorter User Response Time (URT), i.e. the elapsed time between the generation of a request from a browser, for the retrieval of a Web page, and the rendering of that page at that browser's site. Hence, the URT includes both the communication and the processing delays involved in servicing a browser request.

The success of this approach will depend on both the ability (i) to bind dynamically a client to the most convenient replica, and (ii) to maintain data consistency among the WS replicas that implement the same service.

In this paper, we address the issue (i) above. Specifically, we discuss strategies that implement load distribution among WS replicas, distributed over the Internet, so as to minimize the URT. Issues of data consistency among WS replicas fall outside the scope of this paper.

We have assessed both static (e.g. [10, 15]) and adaptive (e.g. [6, 16]) load distribution strategies in distributed systems, in general, as well as specific strategies that have been proposed in order to distribute the processing load among replicated WSs, in particular [1, 2, 3, 4, 5, 9, 11]. For the sake of brevity, we shall not discuss our assessment in this paper. However, we wish to point out that the load distribution strategies for replicated WSs we have examined assume that these WSs are replicated within a cluster of resources, and that the access to them is governed by the cluster DNS server. These strategies aim to maximizing the WSs throughput, rather than minimizing the URT, and pay little attention to such issues as client latency time over the Internet.

We term these strategies "DNS-based" as, in essence, they use the cluster DNS as a scheduler of browsers' requests. However, as discussed in [8], DNS-based load distribution strategies raise problems of caching of browsers' requests (caused by the hierarchical organization of the DNS service), and of maintenance of service availability in the event of host failures (in essence, as the DNS service was designed to support data that infrequently changes, it is not well equipped to propagate changes timely, throughout a system of co-operating servers, if one or more of these servers fail). In addition, their implementation requires, in general, modifications to the DNS server implementation.

Owing to the above observations, in this paper we examine load distribution strategies that require no modifications either to the DNS and to the browsers' and servers' software. Moreover, these strategies deal with WS replicas that can be distributed across the Internet, rather than within a single cluster as in the already cited references.

We consider the following three principal load distribution strategies: a so-called Mirror-based strategy, that provides a browser with access to the particular WS replica that is geographically closer to that browser, a DNS-based strategy, that implements load distribution among WS replicas by using a round robin discipline, and a QoS-based strategy that implements load distribution among WS replicas so as to optimize the URT. A performance comparison of these strategies, that we have carried out, indicates that, in all the scenarios we have considered, the QoS-based strategy outperforms the other two strategies.

This paper is structured as follows. In the next Section we examine the three strategies for load distribution mentioned above, and compare and contrast their relative performance. In

Section 3 we introduce an example architecture (termed "QoS-based architecture" in the following) that implements our QoS-based strategy. Finally, in Section 4, we summarize our work, currently in progress, concerning the definition of alternative QoS-based architectures.

2. LOAD DISTRIBUTION STRATEGIES

In this section we introduce the Mirror-based, the DNS-based, and the QoS-based load-distribution strategies, and compare their performance through a simulation study. The Mirror-based strategy corresponds to that currently used by browsers accessing mirror servers. Specifically, a list of web server replicas is provided to the user who manually selects a replica. Usually the replica server geographically closer to the user site is selected (as no other information is available to the user).

The DNS strategy we assume implements an "ideal" round-robin assignment of clients to WS replicas that maintains a perfect balance of the clients' queries among those replicas (hence the use of the term "ideal" above). Needless to say, this assumption is unrealistic in the actual Internet context. Over the Internet, an address resolution request resolved by a DNS server is cached both in that server and in intermediate name servers, along the path that connects a WS replica to its clients. As pointed out in [8, 4], this caching may cause a deviation from the ideal round-robin scheduling of WS replicas, and may lead to a temporarily unbalanced load distribution among those replicas. However, for the purposes of our discussion, our "ideal" round-robin assumption is useful as it provides us with a convenient benchmark for DNS-based load distribution strategies.

The QoS-based strategy considered in this section makes each browser responsible for implementing its own load distribution. Specifically, in this strategy, it is assumed that the DNS maintains the addresses of all the WSs replicas that implement the available services. When a browser requires access to a service, it interrogates the DNS in order to obtain the addresses of all the replicas that implement that service. Thus, before sending a query to one of these replicas, the browser can probe it (e.g. by sending a *dummy request* to it), in order to assess the URT that replica can provide it with. Eventually, the browser will be able select a replica (if any) that can provide it with a satisfactory URT. Note that, in QoS-based strategy, the probing of replica servers can be implemented by broadcasting UDP messages. Web access, instead, is carried out by using HTTP on top of TCP in the three strategies above.

The QoS-based strategy sketched above requires changes in the DNS implementation, in order to guarantee that a browser obtain all the replica addresses it needs at once; in Section 3, we will show how this strategy can be implemented without introducing any change to the DNS.

2.1 Performance Comparison

In this section we compare the performance of the QoS-based, DNS-based, and Mirror-based load distribution strategies. This comparison is performed via simulation assuming the scenario shown in Figure 1. We consider four replicated Web servers, located in four distinct geographical areas (e.g. USA, Europe, Asia, and Australia). In each area there are local browsers which generate queries for the replicated servers. In our simulation model the time intervals between consecutive queries are independent and exponentially distributed. Each query causes the retrieval of a geometrically distributed number of Web pages. In our simulations a query corresponds to the retrieval of 10 Web pages, in average. Web pages have a size with median 3000 bytes, and a distribution that models the characteristics of the typical HTTP traffic [14].[1] In addition, in the QoS-based strategy case, a dummy request corresponds to the retrieval of a 1000 bytes page. We compare the three load distribution strategies above by analyzing their relative impact on the response time experienced by a query (*Query RT*). Specifically, we estimate both the average and the tail of the *Query RT* distribution. In the QoS-based strategy the *Query RT* includes the delay to select a replica with satisfactory URT. The comparison is carried out assuming several different scenarios. For each scenario, we have run a set of simulative experiments.

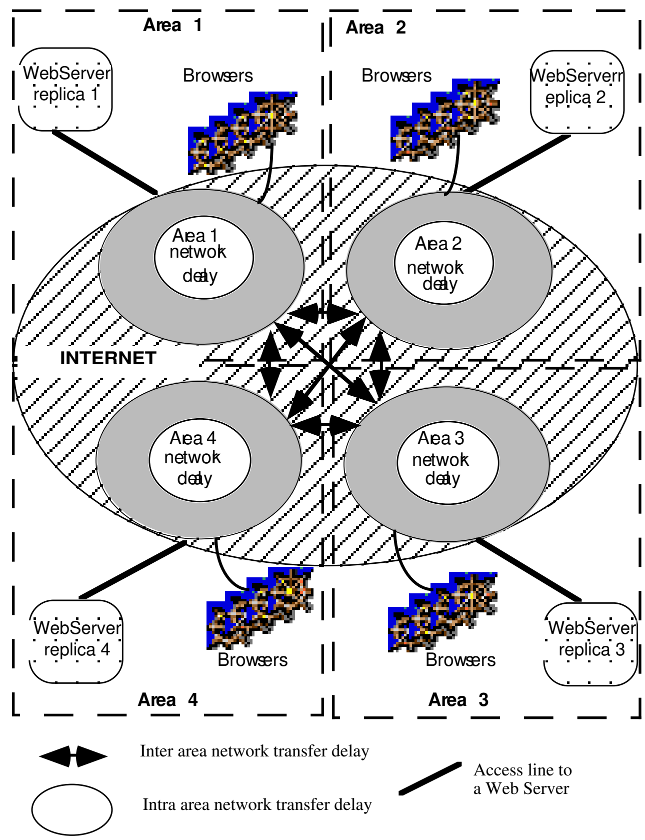

Figure 1: Simulation scenario

The simulation model we have developed can be summarized as follows. The impact of the network congestion on the load distribution strategies is modelled through a set of delays. Specifically, we have modelled two types of network delays: namely, intra-area delays and inter-area delays.

The intra-area delay consists of three components: *i*) the minimum area round trip time (RTT), *ii*) the queuing delays in the area routers, and *iii*) the packet transmission time for each link on the browser-server path.[2]

---

[1] Similar statistics are reported in [13].

[2] In our simulation we assume that the queueing delays in the area routers are primarily caused by a single congested router.

	Area 1 Server	Area 2 Server	Area 3 Server	Area 4 Server
Exp 1	0.58	0.91	0.92	0.92
Exp 2	0.95	0.86	0.85	0.85
Exp 3	0.83	0.83	0.83	0.84
Exp 4	0.44	0.85	0.69	0.51

Table 1: Load distribution with the QoS strategy

	QoS based		DNS based		Mirror based	
	Min	Max	Min	Max	Min	Max
Exp 1	0.92	1.26	1.21	2.39	0.50	1.58
Exp 2	0.61	0.85	1.20	1.33	0.50	2.19
Exp 3	0.57	0.58	1.20	1.23	0.50	0.53
Exp 4	0.62	1.15	1.01	1.95	0.18	4.04

Table 2: Average Query Response Time per Area (seconds)

Inter-area delays are i.i.d. random variables which model the communication latency between different geographical areas. Hence, in our case study, the URT is the sum of the intra-area delays and inter-area delays. Intra-area delays includes queuing delays in the area routers which provide an estimate of the available bandwidth on a link.[3]

In the following simulations the minimum RTT is 10 milliseconds, and the inter-area delays are 50 milliseconds in average.

We assume that each server can serve up to 200 requests per second (*rps*), i.e. the *capacity* of each server is 200 rps. Each server has a finite FIFO queue for buffering incoming queries. Each server is located on an high-speed LAN which is connected to Internet via a router (*access router*). The impact of the LAN on the URT is assumed to be negligible; instead, our model captures the queuing and transmission delays in the access router. The access router has a finite buffer and a tail drop discard policy.

The results presented below have been obtained with the RESQ simulation tool using the independent replication method with a 90% confidence level. The width of the confidence intervals in these results is between 2% and 3%.

2.1.1 Impact of intra-area network congestion (Exp 1)

In this experiment we assume that the areas are balanced from the user-query generation rate standpoint (corresponding to 80% of the servers' capacity), while the network is congested in one area. Specifically, the most congested router in Area 1 has an utilization close to the saturation point (0.98); in the other areas the maximum utilization of the routers is 0.80, instead.

In this scenario, the Mirror-based and the DNS-based strategies produce a uniform distribution of the queries among the WS replicas (80% utilization of each replica). In contrast, the QoS-based strategy tends to underutilize the WS replica in Area 1, as shown in Table 1. This different behaviour can be explained by observing that the intra-area network congestion is transparent to the Mirror-based and the DNS-based strategies.

Instead, the QoS-based strategy takes into consideration the network delays to distribute the queries.[4]

The *Query RT* statistics related to the Area 2 represent the Areas 3 and 4 statistics as well, as these three areas have identical intra-area network delays and user-query generation rate.

By observing both the average and tail of the *Query RT* distribution (see Table 2 [5] and Figure 2), it turns out that:

i) the QoS-based strategy is fairer than the other two strategies. The WS replica chosen for a query does not significantly affect the *Query RT*. This is achieved by compensating the Area 1 network congestion with a lightly loaded WS replica in this area. In fact, as shown in Table 1, the Area 1 WS replica has an utilization equal to 0.58, while the replicas in the other areas have an utilization in the range 0.91-0.92;

ii) the DNS-based strategy exhibits the highest *Query RT* unfairness, as this strategy does not take into consideration the network congestion in Area 1. In particular, the DNS routes one query out of four from each not-congested area to the WS replica in Area 1; hence, one out of four queries experiences both the inter-area communication delays and the intra-area delay of the most congested area.

2.1.2 A heavily loaded area (Exp 2)

In this experiment, we assume that, in Area 1, the user-query generation rate is close to the saturation point of the local WS replica (i.e. 0.98 of the server capacity). In the other areas, the user-query generation rate corresponds to the 80% of the capacity of the local WS replica.

---

[3] The indices we use to identify the minimum URT path are similar to those used in ARPANET routing algorithms to select the shortest path [12].

[4] The QoS-based strategy generates, in each WS replica, an additional load (about 0.03 in our experiments) caused by the *dummy request* used to assess the URT.

[5] The figures reported in Table 2 are calculated, for each strategy, as follows. We compute the average *Query RT* in each area. Among the four values we obtain, we report in the Table the minimum and maximum values.

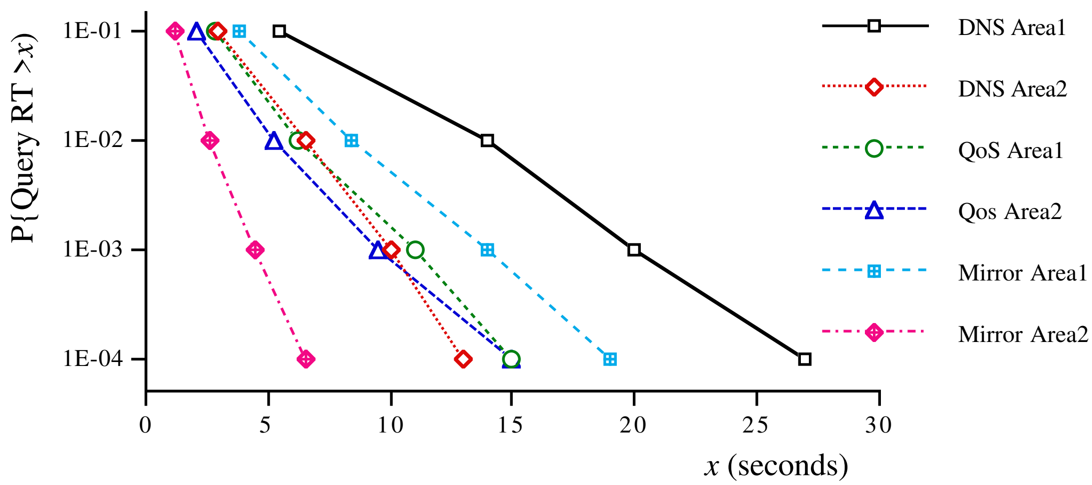

Figure 2: Impact of intra-area network congestion on the query response time

All the four areas are balanced from the network-congestion standpoint (corresponding to an 80% utilization of the most congested router in each area).

In this case, the QoS-based strategy, by redirecting only a small percentage (about 6%) of the Area 1 local queries, produces a fair *Query RT*, i.e. the *Query RT* does not significantly depends on the WS replica used for solving the query.

The Mirror-based strategy does not redirect any Area 1 query; hence, using that strategy, the *Query RT* in Area 1 is much higher than that in the other areas. For example, as shown in Figure 3, the 99th percentile of the *Query RT* is 18 seconds in Area 1, and only 4 seconds in the other areas.

Finally, the DNS-based strategy is fair; however, its query response times are higher than those obtained with the QoS-based strategy. This occurs as the DNS-based strategy tends to redirect about the 75% of the local queries in each area to the WS replicas in the other areas, thus increasing the *Query RT* with the intra-area communication delays.

2.1.3 Symmetric case (Exp 3)

In this experiment, we assume that all areas are balanced both from the network and the server utilization standpoint. Specifically, the most congested router in each area has an utilization equal to 80%, and in each area the user-query generation rate is equal to 0.80 of the local WS capacity. Within this scenario, all the strategies are fair. As it can be expected, in a symmetric case the Mirror-based strategy is the most efficient, as it does not introduce either inter-area communication delays or URT assessment delays (see Figure 4). In this scenario, the DNS-based strategy exhibits the worse performance (see Table 2 and Figure 4), as the *Query RT* in three queries out of four contains also the inter-area communication delays. In contrast, the QoS-based and the Mirror-based strategies exhibit almost identical query response times. This can be explained as the QoS-based strategy, in our experiments, selects a non local WS replica once every 1000 queries, approximately (the small differences with the Mirror-based strategy in the *Query RT* statistics are induced by the URT assessment costs).

2.1.4 A realistic scenario (Exp 4)

We assume that the four distinct geographical areas we consider (i.e. USA, Europe, Asia, and Australia) experience daily different loads in different periods of time (e.g. early morning, rush hour, late afternoon, night). To take into consideration these load differences, we set the intra-area network congestion level and the query-generation rate according to the values reported in Table 3, which relates to one particular period of the day.

Figure 5 plots the tail of the *Query RT* distribution related to Area 1 (the most congested) and to Area 4 (the least congested). In this experiment, both the DNS-based and the Mirror-based strategies are extremely unfair. The former only balances the load among the WS replicas, and does not consider the impact of the Area 1 network congestion. The latter, as it does not redirect the queries generated in an area towards the WS replicas in other areas, produces the highest *Query RT* unfairness. Specifically, this strategy tends to saturate the Area 1 WS replica, causing large queuing delays before a query can be processed by this replica. In addition, queries addressed to the Area 1 WS replica experience the largest intra-area network delays. Moreover, this strategy causes underutilization of the Area 4 WS replica. Hence, the Area 4 local queries do not experience queuing delays both in the routers and in the WS replica. In contrast, the QoS-based strategy redirects almost 50% of the Area 1 queries to the lightly loaded Areas 3 and 4. Specifically, our simulation results show that the Area 4 WS replica has an utilization of 51% (see Table 1), and a local query rate of 0.10. This means that most of the queries (0.38) to the Area 4 WS replica originate from Area 1.

3. A QOS-BASED ARCHITECTURE

The performance comparison of the three load distribution strategies discussed in the previous Section indicates that, in all scenarios we have considered, the QoS-based strategy outperforms the other two strategies. However, as pointed out earlier, the QoS-based strategy we have assumed requires changes to the DNS.

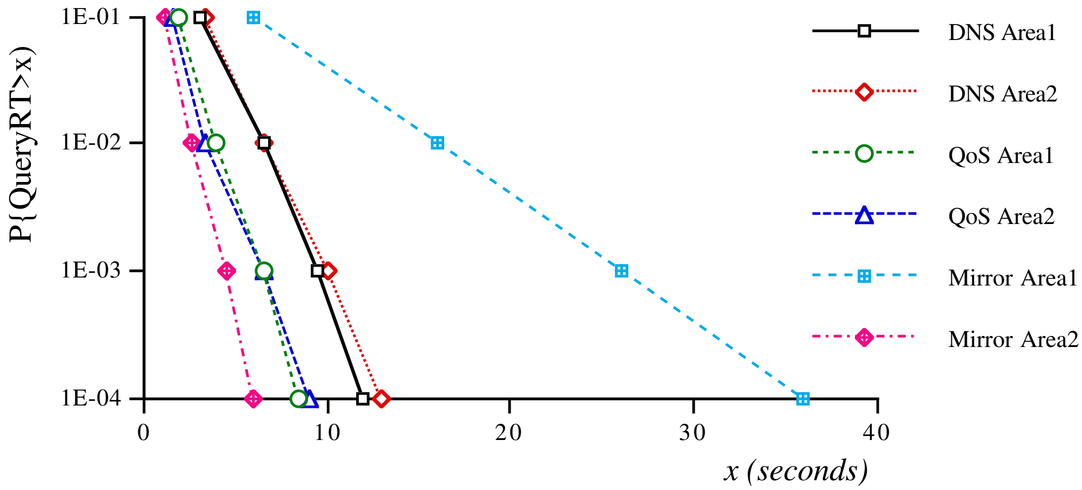

Figure 3: Impact of a heavily loaded area on the query response time

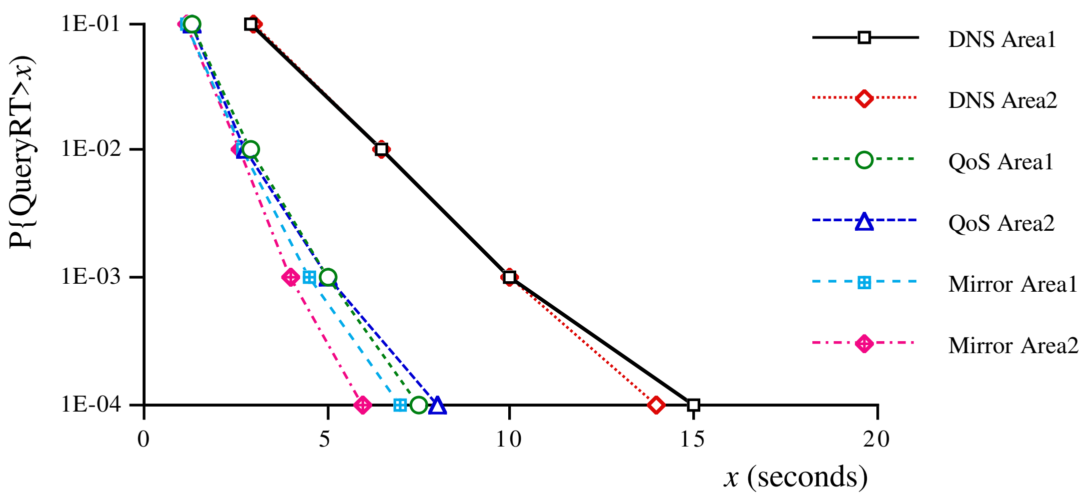

Figure 4: Query response time in a symmetric case

The architecture we present below implements that QoS-based strategy, and does not require modifications either to the DNS or to the browsers' and servers' software. As we assume that a replicated Web service is implemented by WS replicas that are distributed across the Internet, as depicted in Figure 6, each replica is addressed via a different Internet address.

The architecture we propose embodies two principal components: namely, a Web server Software Layer (WSL), and a Browser Software Layer (BSL). The former implements the QoS-based strategy at the server side, and is implemented on top of the TCP/IP interface in each replica server machine; the latter is responsible for implementing the QoS-based strategy at the browser side, by co-operating with the WSL in each WS replica, and is implemented on top of the TCP/IP interface in each browser machine.

Both the BSL and the WSL software layers can be constructed so as to maintain the standard TCP/IP interface (see Figure 7); thus, the implementation of higher level software, such as the HTTP and the browser and server code, can operate unaltered.

A straightforward implementation of the QoS-based strategy presented in Section 2, which does not require any change to the DNS, can be summarized as follows. Each WS replica maintains the addresses of its peer WS replicas that implement the same replicated service. When a browser requires to access one such service, it interrogates the DNS in order to obtain the address of a WS replica. Then, it issues a poll message to that replica.

The replica returns to the browser both a reply to the poll message and the addresses of its peer replicas. If the browser deems acceptable the URT resulting from the polling of that replica, it selects that replica for servicing its Web requests. Otherwise, the browser broadcasts a poll message to the other replicas, in order to assess the URT they can provide, and selects the replica characterised by the shorter URT.

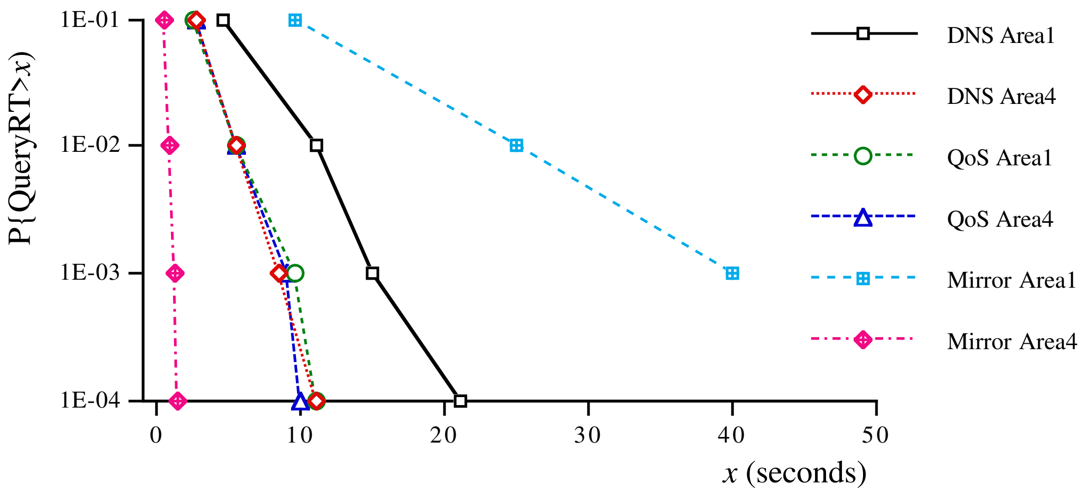

Figure 5: Query response time in the *realistic scenario*

	Area 1	Area 2	Area 3	Area 4
network	0.98	0.80	0.50	0.10
query rate	0.98	0.80	0.50	0.10

Table 3: Load configuration for the realistic scenario

This approach is attractive as it does not require either modifications to the DNS, or the implementation of a specific load distribution service (in particular, the additional code for load distribution may well be incorporated in the browser configurable cache, as this feature is available in most commercial browsers). However, this approach has some drawbacks. The most critical drawback is related to the URT estimation process which is based on a single measure. It is worth noting that the URT is the sum of two components: the network round-trip time (URT_{RTT}) and the WS response time (URT_{WS}). The URT_{RTT} component is almost independent of the load distribution strategy one uses, as the amount of traffic affected by that strategy is a small part of the aggregate network traffic. Instead, the URT_{WS} is strongly correlated with traffic spreading induced by the load distribution strategy (and vice versa). Hence, the URT_{WS} estimates based on a single measure may introduce harmful fluctuations of the Web queries among the WS replicas.

An additional drawback of this architecture is the polling overhead. In fact, polling may cause that WS replicas be overloaded with a number of requests of URT assessment that is proportional to the number of service requests that the browsers issue.

4. WORK IN PROGRESS

To overcome the main limitations of the architecture introduced above, we require a more reliable, and possibly less costly, method to perform the URT assessment. To this end, we are currently investigating several architectures which, in addition to the BSL and WSL, make use of a further architectural component, termed Load Distribution (LD) service. Specifically, a LD service is responsible for distributing the browsers' requests, directed to a (redundant) Web service, to the appropriate WS replica that implements it.

The principal responsibility of the LD service is to maintain for each WS replica the estimate of its URT_{WS}. To this end, the LD service can perform a continuous monitoring of the response time each replica can provide. In this way we can solve the problems, pointed out in the previous section, related to the URT_{WS} estimates based on a single measure.

Different architectural approaches can be used to implement a LD service. For example, that service can be implemented by a **centralized server**. However, an architecture based on one such server is extremely vulnerable to congestion; in addition, it introduces a single point of failure in the Web system that uses that LD server.

In order to overcome these problems, the centralized server can be replaced by a distributed implementation of the LD service, as proposed below.

The Internet introduces the concept of Autonomous System (AS), that is "a region of the Internet under the administrative control of a single entity" [12]. In practice, an AS can consist of a collection of networks and routers managed by one administrative authority. Based on this concept, a LD service can be implemented according to one of the following two alternative approaches.

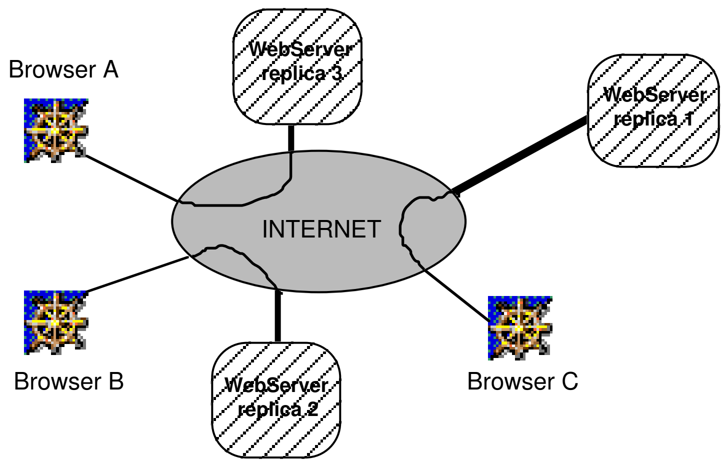

Figure 6: Replicated Web service

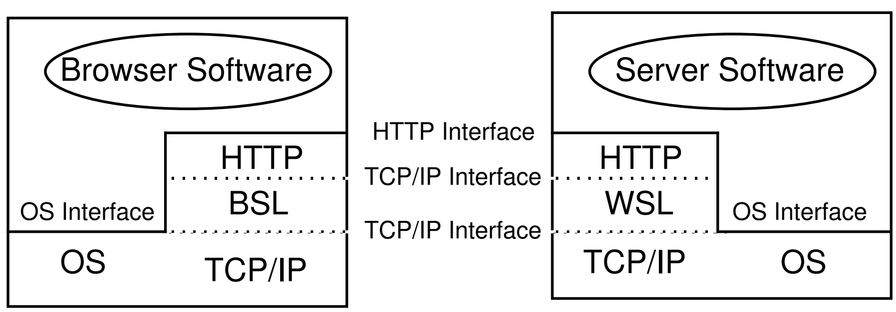

Figure 7: Example of browser and server software structuring.

LD Server in the AS of each WS replica. A service provider (e.g. a corporation), that makes available a replicated service, can effectively control a number of geographically dispersed ASs. Hence, a LD server instance can be associated to each AS belonging to that service provider, and WSs can be replicated in these ASs. In essence, a LD server can be physically located at the boundary between an AS, incorporating replica servers, and the rest of the Internet (e.g. in the border router). Each LD server will be responsible for managing the load among the WS replicas in the AS it belongs to.

LD Server in the AS of the Browsers. A LD server can be instantiated in the AS of each browser. This approach would allow the LD servers to estimate, reasonably accurately, and possibly off-line, the costs of communications between the browsers and the various WS replicas. However, this approach would lead to an architecture that hardly scales as, at least in principle, each AS in the Internet should incorporate a LD server.

In conclusion, we wish to point out that we will compare and contrast these architectures by adopting simulation and measurements. As far as the simulation is concerned we will adopt real traces of the Internet traffic for load characterization purposes [1].

5. ACKNOWLEDGMENTS

We wish to thank the WISP'99 anonymous referees for their valuable comments.

6. REFERENCES

[1] R.B. Bunt, D.L. Eager, G.M. Oster, C.L. Williamson, "Achieving Load Balance and Effective Caching in Clustered Web Servers", Proc. *4th International Web Caching Workshop*, San Diego, CA, March 1999.

[2] "Cisco LocalDirector", CISCO Systems Inc., White paper, 1996.

[3] M. Colajanni, P. S. Yu, D. M. Dias, "Analysis of Task Assignment Policies in Scalable Distributed Web-Server Systems", IEEE Trans. on Parallel and Distributed Systems, Vol. 9, NO. 6, pp. 585-600, June 1998.

[3] M. Colajanni, P. S. Yu, V. Cardellini, "Dynamic Load Balancing in Geographically Distributed Heterogeneous Web Servers", Proc. IEEE 18th Int. Conf. on Distributed Computing Systems (ICDCS'98), Amsterdam (The Netherlands), pp. 295-302, May 1998.

[5] O P. Damani, P.E. Chung, Y.Huang, C.Kintala, Y.-M. Wang, "ONE-IP: Techniques for Hosting a Service on

aCluster of Machines", *Computer Networks and ISDN Systems*, 29, 1997, pp. 1019-1027.

[6] D.L. Eager, E.D. Lazowska, J. Zahorian, "Adaptive Load Sharing in Homogeneous Distributed Systems", IEEE Trans. on Software Engineering, Vol. SE-12, NO.5, pp. 662-675, May 1986.

[7] M.J. Fisher, N.A. Linch, M.S. Paterson, "Impossibility of Distributed Consensus with One Faulty Process", Journal of the ACM, Vol. 32, NO. 2, pp. 374-382.

[8] D. Ingham, F. Panzieri, S.K. Shrivastava, "Constructing Dependable Web Services", Proc. Joint International Computers Limited/University of Newcastle upon Tyne Seminar "THE WEB", B. Randell (Ed.), 1 - 5 Sept. 1997, Newcastle upon Tyne (G.B.), pp. VIII.3 - VIII.17.

[9] E.D. Katz, M. Butler, R. McGrath, "A Scalable HTTP Server: The NCSA Prototype", Comp. Net. and ISDN Sys., 27 (2), pp.155-164, Nov. 1994.

[10] J. Li, H. Kameda, "Load Balancing Problems for Multiclass Jobs in Distributed/Parallel Computer Systems", IEEE Trans. on Computers, Vol. 47, NO. 3, pp. 322-332, March 1998.

[11] V. Pai, G. Banga, M. Svendsen, P. Druschel, W. Zwaenepoel, E. Nahum,"Locality-aware Request Distribution in Cluster-based Network Servers", Proc. *ASPLOS- VIII*, San Jose, CA, October 1998.

[12] L.L. Peterson, B.S. Davie, *Computer Networks*, Morgan Kaufmann, San Francisco, CA, 1996

[13] J.E. Pitkow, "Summary of WWW Characterizations', *Computer Networks and ISDN Systems*, 30, 1998, pp. 551-558.

[14] W.R.Stevens, *TCP/IP Illustrated, Volume 3*, Addison-Wesley, Reading, MA, 1996

[15] A.N. Tantawi, D. Towsley, "Optimal Static Load Balancing in Distributed Computer Systems", J. ACM, Vol. 32, NO.2, pp. 445-465, Apr. 1985.

[16] J. Watts, S. Taylor, "A Practical Approach to Dynamic Load Balancing", IEEE Trans. on Parallel and Distributed Systems, Vol. 9, NO. 3, March 1998, pp. 235-248.