An Overview of Data Replication on the Internet

Thanasis Loukopoulos, Ishfaq Ahmad* and Dimitris Papadias

Department of Computer Science
The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong

*Department of Computer Science and Engineering
The University of Texas at Arlington, Arlington, Texas, USA

Abstract

The proliferation of the Internet is leading to high expectation on the fast turnaround time. Clients abandoning their connections due to excessive downloading delays translates directly to profit losses. Hence, minimizing the latency perceived by end-users has become the primary performance objective compared to more traditional issues, such as server utilization. The two promising techniques to improve the Internet responsiveness are caching and replication. In this paper we present an overview of recent research in replication. We begin by arguing on the important role of replication in decreasing client perceived response time and proceed by illustrating the main topics that affect its successful deployment on the Internet. We analyze and characterize existing research, providing taxonomies and classifications whenever possible. Our discussion reveals several open problems and research directions.

1 Introduction

The World-Wide Web [10] has been established as the de facto source of information for common users, and is now responsible for the majority of Internet traffic. E-commerce, e-banking and www publishing are areas that generate an ever increasing interest for web applications. The boost in the popularity of the web resulted in large bandwidth demands and notoriously high latencies experienced by end-users. To combat these problems the benefits of caching and replication were early recognized by the research community.

Caching [11] was traditionally applied to distributed file systems such as the AFS [48] and although it is a well studied issue, its application on the Internet gave rise to new problems, e.g., where to place a cache, how to make sure cached contents are valid, how to handle dynamic pages etc. Replication was commonly used in distributed systems to increase availability and fault tolerance. Both techniques have complimentary roles in the web environment. Caching attempts to store the most commonly accessed objects as close to the clients as possible, while replication distributes a site's contents across multiple mirror servers. Caching targets directly at minimizing the download delays, by assuming that retrieving the required object from the cache, incurs less latency compared to getting it from the web server. Replication, on the other hand, accounts for improved end-to-end responsiveness by allowing clients to perform downloads from their closest mirror server (either in geographic terms or network-wise), while at the same time balancing the load among the various web servers.

From a terminology point of view, caching and replication are sometimes used interchangeably in the literature. Arguably, caching can be viewed as a special case of replication when mirror servers store only parts of a site's contents. This analogy leads to some interesting comparisons. For instance, cache replacement algorithms are examples of on-line, distributed, locally greedy algorithms for data allocation in replicated systems [26]. Furthermore, the fact that caches do not have full server capabilities can be viewed as the case of a redirection policy for a replicated system, that sends requests for specific object types (e.g., dynamic pages) to a single server. Essentially, every major aspect of a caching scheme has its equivalent in replicated systems, but not a vice versa.

Until now, researchers concentrated their efforts primarily on caching issues and the potential of replication was not fully exploited. Therefore it comes as no surprise that the current state of the art involves manual decisions concerning where to create a mirror site and copying the whole site's contents (coarse grain replication) even if some objects are rarely accessed. Moreover, requiring the clients to explicitly define the server from where to download, is still the order of the day for many sites.

In this survey we attempt to: 1) provide an overview of the key issues when applying replication on the web, 2) summarize and categorize research done in the area and 3) identify open problems and directions for future work. A survey on web caching can be found in [59]. A number of bibliographies and reading materials for web caching are also available on line, e.g., [23]. Concerning replication and its challenges, [53] provides an early overview.

The rest of the paper is organized as follows. Section 2 focuses on major research issues in replication. Section 3 discusses redirection. Section 4 discusses server selection and Section 5 describes issues in replica placement. We illustrate some open research problems in Section 6. Finally, Section 7 presents some concluding remarks.

2 Major Issues in Replication

The large volume of requests (potentially millions per day) arriving at popular sites, can result in saturating even the most powerful web server, especially during peak hours. In order to tackle the problem, hot sites run multiple servers, potentially spanning the globe. The objective of

Proceedings of the International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN'02)
1087-4089/02 $17.00 © 2002 **IEEE**

replication is to make all these servers appear to the user as a single highly capable web server.

Replication began by manually mirroring web sites. The current target is to dynamically create, migrate and delete replicas among web hosts in response to changes in the access patterns. The demand for dynamic replication comes from the continuous increase in the large scale web hosting market [50], where it is evident that manual manipulation of a huge number of replicas is not feasible.

There are three main issues in implementing a replicated service on the Internet. The first is to assign requests to servers according to some performance criterion in a transparent to the end user way. This in turn gives rise to two new problems namely, who should decide about the request redirection (location) and where should a request be directed (server selection). The second challenge is to decide on the number and placement of replicas across the distributed servers. Last but not least, is the problem of maintaining consistency in the face of updates. In this survey, we leave aside consistency issues and focus on the rest.

3 Location of Redirection

Depending upon where the redirection occurs, the various schemes proposed in the literature could fall under the following main categories:

Client Side Redirection: The web browser obtains a set of physical replicas and chooses where to send the request. This approach is used for example in [7] and [59]. The scheme in [59] relies on Java applets carrying knowledge of the actual replication scheme. URLs point to these applets and the choice is done at the client's side. Although such policy scales well, the overhead for fetching and running the applet can not be neglected. In [7] information about replica sets is propagated in HTTP headers. The scheme although scalable, requires changes to be made at both servers and clients in order to process the headers.

Router Redirection: A multiplexing router is placed in front of a server farm and the domain name of the web site is mapped to the IP address of the router. Upon receiving the first packet from a client the router selects a server in the farm and forwards the request. A database for the active sessions is kept in order to make sure that all packets from a single client end at the same server. Server respond packets are intercepted and their headers are modified so as to contain the router's IP address. CISCO's Local Director [20] implement this policy.

DNS Redirection: Several schemes proposed in the literature take advantage of the DNS [46] infrastructure to redirect requests. In [8], the DNS server of the service provider returns the set of IP addresses corresponding to the locations of different replicas. The DNS resolver at the client side decides where to send the request, possibly after probing the servers and measuring the response time. [19] also uses the DNS backbone for redirection only that server selection is done at the sites' DNS server, with the aim of load balancing the servers. Although, DNS based redirection adds no extra communication on the critical path of request processing it suffers from another disadvantage. The performance of the DNS backbone, relies heavily on caching recently resolved names, which can result in stale replica selection and server overloading, thus hindering the scalability and performance of DNS based redirection schemes.

In [55] the authors quantify the impact (in terms of latency) of reducing the TTL values of the cached name resolutions. They provide empirical evidence that the latency can rise by two orders of magnitude when DNS resolution is done without caching. Moreover, trace analysis of dial-up ISP clients indicated that clients can be very distant from their corresponding nameservers, which can lead to redirecting requests to suboptimal servers. [60] studies the contribution of DNS lookups when retrieving web documents. Simulation results showed that the DNS entries which return multiple IP addresses, change more frequently compared to those returning a single one.

Server Side Redirection: Some of the proposed techniques rely on the web servers performing request redirection. Examples of this approach are the SWEB system [3] and the Distributed Packet Rewriting (DPR) [12]. In SWEB each server decides whether to serve a request locally or redirect it using the HTTP redirection functionality [62]. The decision is taken based on the current workload exhibited by the servers, which requires the servers to periodically broadcast their state. DPR on the other hand, uses a packet rewriting mechanism which allows a web server to act essentially as a router, redirecting the connection elsewhere. Implementation details on distributed connection routing using DPR can be found in [5]. A positive aspect of server side redirection is the fact that it allows the actual content of a request to determine the redirection decision (e.g., a server optimized to run CGI scripts should satisfy the biggest portion of such requests).

4 Server Selection

Various methods for selecting among replicated servers are extensively discussed in the relevant literature, most of them attempting to provide better QoS by assigning a client to its "closest" server. The response time a user experiences depends on: i) server capabilities, ii) server load, iii) network path characteristics, e.g., propagation delay, iv) path load. The solutions proposed vary according to how many factors they consider. [2] and [16] provide an overview of some of the server selection alternatives proposed in the literature and discuss the expected benefits and drawbacks.

In [22] the authors argue that *hop* distance is not a good metric. They examine the performance of a *random* algorithm, that selects servers uniformly, as well as one which selects the server with the minimum *average* response time. Experiments show that dynamic server selection in the form of the above algorithms outperforms static client-server allocation. Another study [54] examines the performance of different client-server distance metrics. The correlation between the number of hops and the response time, is found to be low (0.16), supporting the results of [22]. The use of *ping* utility as a metric is also deemed unsatisfactory (correlation of 0.51). The best metric is found to be the *past request latency*. Using past history as a metric, the authors propose a *probabilistic* and a *refresh* algorithm. The probabilistic selects a server with a probability inversely proportional to the latency exhibited in the past, thus not leading to starvation as a straightforward client allocation would. The refresh algorithm asynchronously updates latency information by sending periodically an HTTP HEAD request to the corresponding server, thus avoiding suboptimal client-server allocations due to latency information being outdated. Both algorithms were marginally outperformed by a *parallel* strategy, where a client sends his request to all the available servers and selects the one who responds first. In [33] the authors experiment with an algorithm based on calculating a *weighted average* of past performance so as to place more importance on recent measurements. Their approach compares favorably to traditional schemes, such as *round robin*, but is dependent on careful parameter selection. The study in [35] also considers latency as the distance metric. It examines various methods for latency computation and compares different server selection algorithms, using their execution time as the primary performance criterion.

The above techniques are useful primarily when redirection is client based. If this is not the case, distance estimation becomes more complicated and inaccurate. Let us consider server based redirection as an example. When a request arrives at a server it is possible to probe the client and measure the response time. The inter-server distances are also computable. Nevertheless, it is not always possible to determine with accuracy the closest server to the client due to the triangulation phenomenon. An early work in [34] describes how to derive client-server distance metrics and proposes techniques based on polling routing tables and network probing (i.e., placing measuring agents at key points in the Internet and calculate their distances). SONAR [47] and IDMaps [31] aim at building and maintaining an infrastructure on the Internet, capable of returning the distance between any pair of hosts. SONAR describes a DNS-like infrastructure to support client queries, while IDMaps tackles the problem of how to calculate and maintain the actual distance metrics. The method they propose is based on clustering of hosts and approximating distances using triangulation. The work in [56] is related to DNS based redirection. The authors propose extensions to routers and DNS servers in order to allow the later to communicate and take advantage of the reliable distance information kept in the routing tables [25].

A slightly different problem is addressed in [30], i.e., how to maintain the metric information needed by the replicated servers. They consider three strategies: i) Server push. Upon the occurrence of significant metric changes a server notifies the other replica holders. It incurs minimum overhead, but is inaccurate since it does not measure path characteristics; ii) Client probing. Periodically probe agents issue queries to the servers and measure the response time; iii) A Hybrid server push/client probing approach, which was experimentally found to be the most promising.

Server selection is also examined from the perspective of load balancing a cluster of web servers instead of minimizing end to end response time. The study in [21] provides a set of scheduling algorithms when redirection happens at the DNS level. It is observed that algorithms using asynchronous “server overloaded” warning messages, in combination with characteristics of client origin, perform significantly better compared to algorithms maintaining the accurate workload state of the servers. The same authors in [15] provide extensive simulation results and compare the most successful methods of [21], with another alternative, namely Adaptive TTL. The new method, which is based on deriving optimal TTL values for DNS responses, was found to increase system’s performance. Finally, [14] studies scheduling techniques that involve DNS - web server cooperation, with the redirection decisions occurring at the web servers using HTTP functionalities.

5 Replica Placement

The question of which replicas to place where in order to increase the overall system performance, received surprisingly little attention by the Internet community, presumably due to the fact that most research efforts focused on optimizing caching mechanisms. Nevertheless, with the recent deployment of large content distribution networks (Inktomi [36], Exodus [29], Digital Island [24]) that offer hosting services to multiple content providers, allocation issues become more and more important. There exist a number of problem definitions and algorithms, each one attempting to characterize and improve different performance aspects. Following, we classify research efforts in the area depending on their proximity with well known theoretical problems[1].

k-Median [45]: A description of the k-Median problem, which is NP-hard, is as follows: We are given a graph *G(V, E)* with weights on the nodes representing number of requests and lengths on the edges. Satisfying a request, incurs network cost equal to the length of the shortest path between the origin node and a server. The problem consists of placing *k* servers on the nodes, in order to minimize the total network cost, given that each node can hold at most one server.

The k-Median formulation is used in order to tackle with the problem of distributing a single replica over a fixed number of hosts. Most papers in this category assume a replica to be a mirror server hosting all site contents, thus performing coarse grain replication. In [41] the authors study the problem of placing *M* proxies at *N* nodes when the topology of the network is a tree and proposed an $O(N^3M^2)$ algorithm that finds the best solution. The same authors in [58] reduce the complexity of their approach. Although both papers are significant from a theoretic point of view the tree network assumption reduces the applicability of their approach in real cases.

1. Some of the referred cited papers were written before the advent of the Internet. They are cited here to provide a more comprehensive picture.

The work in [51] tackles the problem of replica placement in order to reduce network bandwidth consumption. They provide a greedy heuristic that outperforms the dynamic programming method of [41] in the non-tree network case. The study in [38] investigates the optimal placement of Internet distance measuring instrumentations under the IDMaps framework [31]. Both graph theoretic and heuristic algorithms are examined with results varying depending on the network topology. The performance metric considered is the expected accuracy of distance estimations, produced under the different instrumentation assignments. In a more recent work [39], the same authors compared a 2-approximation algorithm for the k-Median, with a greedy approach, a random algorithm and a heuristic which favoured nodes of higher outdegree for replica placement.

Bin packing [32]: An informal description of the bin packing problem is as follows: Given *N* objects, of various sizes, partition them to the minimum number of disjoint sets such as the cumulative storage at each set does not exceed a threshold *S*. Bin packing formulation is commonly used to model load balancing problems. The problem of distributing documents in a cluster of web servers in order to balance their load is discussed in [49]. The paper proposes a binning algorithm for the initial distribution and network flow [1] formulations in the case where either access patterns change or there is a server failure. In order to achieve graceful performance degradation dummy-replicas are also distributed (i.e., replicas that satisfy requests only in case the primary replica fails).

File Allocation [26]: The basic form of the file allocation problem (FAP) is the following: Given a network of *M* sites with different storage capacities and *N* files exhibiting various read frequencies from each site, allocate the objects to the sites in order to optimize a performance parameter (e.g., total network traffic), with respect to the storage capacity available at each node [17]. The study in [18] expands the formulation to account for multiple object copies and updates, while [28] proves that FAP is NP-complete. In [44] the authors provide an iterative approach that achieves good solution quality when solving the FAP for infinite site capacities. A complete although old survey on the FAP can be found in [26]. FAP originated from the need to allocate programs and data to multiprocessor systems [17] and in its general formulation can be viewed as a case of the uncapacitated facility location (UFL) problem [45] which is studied in the business management sector. Subsequently FAP-like formulations were used to describe similar problems arising in distributed databases [4], [9], multimedia databases [40] and video server systems [13], [42].

Most of the papers included in [26] account for static centralized allocation (i.e., access patterns are assumed to be known and remain unchangeable). The Internet environment poses additional challenges with its dynamic nature. The work in [43] uses genetic algorithms to solve both the static version of the problem and an adaptive version whereby changes on object popularities make the current replica distribution obsolete and a fast way to determine a new allocation scheme is needed. Taking into account reliability and availability constraints is the focus of [57] where the authors combine allocation together with fault tolerance issues in a single formulation.

The studies in [6], [52] and [61] are concerned with the on-line version of FAP, i.e., they propose algorithms which given an input stream of requests, alter replica distribution so as to minimize the total answering cost. The problem can be viewed as an extension to the well studied paging problem (see [37] for a survey). In [61] the authors proposed a protocol called ADR that changes the replication scheme of an object so as to minimize the network traffic due to reads and updates. To do so, they impose a logical tree structure for the network. Authors in [52] propose a general protocol that creates, migrates and deletes replicas so as to improve the client-replica proximity without overloading any of the servers. The model does not account for updates. The work reported in [6] is of significant theoretical value providing an optimally competitive centralized algorithm as well as a decentralized one to solve the on-line FAP. .

6 Open Research Issues

Research consensus indicates that DNS based redirection ensures maximum scalability and that the best client-replica proximity metric is the average past latency or a weighted variant of it. Furthermore, concerning the load balancing issue, significant work in order to alleviate the negative impact of cached DNS responses has been reported. Building a host distance map is yet another hot topic that is attracting research efforts. However, to the best of our knowledge the issue of finding suitable tradeoffs between selecting always the closest server to redirect a request and always attempting to balance the load has not yet been addressed. The silent consensus is that at a global level the performance metric should be proximity, while at a local level (i.e., within a server cluster) redirection should be done with respect to load balancing. We believe that better solutions for the global level are possible. On a related issue little research was done concerning how to combine redirection mechanisms (especially those who aim at load balancing) with replica placement.

The research for the replica placement problem is focused on k-Median formulations and coarse grain replication. The FAP formulations are very suitable to describe the problem of distributing the contents of large CDNs, since they formulate the placement of multiple sites and take into account storage space limitations. Dynamic allocation provides various challenges, since excessive page migrations can offset any benefits and is still open for research. Another issue not adequately addressed yet is how, starting from a starting replication scheme, can we schedule a series of replica transfers in order to realize a new replication scheme, presumably determined by either a k-Median or a FAP formulation. Finally, little work is done on characterizing the various tradeoffs between coarse grain/per site and fine grain/per page replication. Although it is straightforward that per page distribution accounts for better resource usage, it places significant additional burden to the DNS (which needs to keep per page entries instead of per site), while increasing the complexity of the placement algorithms. It is likely that grouping pages according to their popularity and considering the composite objects as the grain for replication can offer a better tradeoff.

7 Conclusions

In this paper we studied the major challenges posed from applying replication to increase Internet performance. We analyzed and summarized research output in each of the areas, while describing open problems and avenues for future research. The main contribution of our work is related to the extensive and systematic approach we followed. This is especially true concerning replication, a topic we believe will be of ever increasing interest.

References

[1] R. Ahuja, T. Magnanti and J. Orlin, "Network Flows: Theory, Algorithms, and Applications," *Prentice Hall*, 1993.

[2] Y. Amir, A. Peterson, and D. Shaw, "Seamlessly Selecting the Best Copy from Internet-Wide Replicated Web Servers," *in Proc. of the 12th Int. Symposium on Distributed Computing*, Andros, Greece, Sept. 1998.

[3] D. Andresen, T. Yang, V. Hohmedahl and O. Ibarra, "SWEB: Toward a Scalable World Wide Web Server on Multicomputers," *in Proc. of the 10th Int. Symp. on Parallel Processing*, Honolulu, pp. 850-856, April 1996.

[4] P.M.G. Apers, "Data Allocation in Distributed Database Systems," *ACM Trans. on Database Systems*, 13(3), Sep. 1988, pp. 263-304.

[5] L. Aversa and A. Bestavros, "Load Balancing a Cluster of Web Servers Using Distributed Packet Rewriting," *in Proc. of the IEEE Int. Performance, Computing, and Communication Conference (IPCCC'2000)*, Phoenix, AZ, February 2000.

[6] B. Awerbuch, Y. Bartal and A. Fiat, "Optimally-Competitive Distributed File Allocation," *in Proc. of the 25th Annual ACM STOC*, Victoria, B.C., Canada, 1993, pp. 164-173.

[7] M. Baentsch, L. Baum, G. Molter, S. Rothkugel and P. Sturm, "Enhancing the Web Infrastructure - from Caching to Replication," *IEEE Internet Computing*, pp. 18-27, Mar-Apr. 1997.

[8] M. Beck and T. Moore, "The Internet-2 Distributed Storage Infrastructure Project: An Architecture for Internet Content Channels," *in Proc. of the 3rd Int. WWW Caching Workshop*, Manchester, UK, June 1998.

[9] L. Bellatreche, K. Karlapalem and L. Qing, "An Iterative Approach for Rules and Data Allocation in Distributed Deductive Database Systems," *in Proc. of the 7th International Conference on Information and Knowledge Management (ACM CIKM'98)*, pp. 356-363, November 1998, Washington D.C.

[10] T. Berners-Lee, R. Cailliau, A. Luotonen, H. Nielsen and A. Secret, "The World-Wide Web," *Communications of the ACM*, vol. 37, no. 8, pp. 76-82, Aug. 1993.

[11] A. Bestavros, "WWW Traffic Reduction and Load Balancing through Server-based Caching," *IEEE Concurrency: Special Issue on Parallel and Distributed Technology*, Vol.5, pp. 56-67, Jan.-Mar. 1997.

[12] A. Bestavros, M. Crovella, J. Liu and D. Martin, "Distributed Packet Rewriting and its Application to Scalable Server Architectures," *in Proc. of the 6th Int. Conference on Network Protocols (ICNP'98)*, Austin, Texas, Oct. 1998.

[13] C. Bisdikian and B. Patel, "Cost-Based Program Allocation For Distributed Multimedia-on-Demand Systems," *IEEE Multimedia*, 3(3), pp. 62-72, 1996.

[14] V. Cardellini, M. Colajanni and P. Yu, "Redirection Algorithms for Load Sharing in Distributed Web-Server Systems," *in Proc. of the 19th IEEE Int. Conf. on Distributed Computing Systems ICDCS'99*, Austin, Texas, May 1999.

[15] V. Cardellini, M. Colajanni and P. Yu, "DNS Dispatching Algorithms with State Estimators for Scalable Web-Server Clusters," *World Wide Web Journal*, Baltzer Science, Vol. 2, No. 3, pp. 101-113, Aug. 1999.

[16] V. Cardellini, M. Colajanni and P. Yu, "High Performance Web-Server Systems," *in Proc. of the 13th Int. Symp. On Computer and Information Sciences (ISCIS'98)*, Ankara, Turkey, pp. 288-293, Oct. 1998.

[17] R. Casey, "Allocation of Copies of a File in an Information Network," *in Proc. of the Spring Joint Computer Conf.*, IFIPS, 1972, pp. 617-625.

[18] W. Chu, "Optimal File Allocation in a Multiple Computer System," *IEEE Trans. Computers*, vol. C-18, no. 10, 1969.

[19] CISCO, "Distributed Director," White paper at: http://www.cisco.com/warp/public/cc/cisco/mkt/scale/distr/tech/dd_wp.htm.

[20] CISCO, "Scaling the Internet Web Servers," White paper at: http://www.cisco.com/warp/public/cc/cisco/mkt/scale/locald/tech/scale_wp.htm

[21] M. Colajanni, P. Yu and D. Dias, "Scheduling Algorithms for Distributed Web Servers," *in Proc. of the 17th IEEE Int. Conf. On Distributed Computing Systems (ICDCS'97)*, May 1997.

[22] M. Crovella and R. Carter, "Dynamic Server Selection in the Internet," *in Proc. of the 3rd IEEE Workshop on the Architecture and Implementation of High Performance Communication Subsystems (HPCS'95)*, June 1995.

[23] B. Davison's Web Caching Bibliography, available at: http://www.cs.rutgers.edu/~davison/web-caching/bibliography.html

[24] Digital Island, available at: http://www.digitalisland.com.

[25] J. Doyle, "Routing in TCP/IP volume I," *Macmillan Technical Publishing*, first edition, 1998.

[26] L.W. Dowdy and D.V. Foster, "Comparative Models of the File Assignment problem," *ACM Computing Surveys*, Vol.14(2), June 1982.

[27] V. Duvvuri, P. Shenoy and R. Tewari, "Adaptive Leases: A Strong Consistency Mechanism for the World Wide Web," *in Proc. of the IEEE INFOCOM'99 Conf.*, 1999.

[28] K.P. Eswaran, "Placement of Records in a File and File

Allocation in a Computer Network," *Information Processing*, 1974, pp. 304-307.

[29] Exodus, available at: http://www.exodus.com.

[30] Z. Fei, S. Bhattacharjee, E. Zegura, and M. Ammar, "A Novel Server Selection Technique for Improving the Response Time of a Replicated Service," *in Proc. of the IEEE INFOCOM '98 Conf.*, March 1998.

[31] P. Francis, S. Jamin, V. Paxson, L. Zhang, D. Gryniewicz, and Y. Jin, "An Architecture for a Global Internet Host Distance Estimation Service," *in Proc. of the IEEE INFOCOM '99 Conf.*, March 1999.

[32] M. Garey and D. Johnson, "Computers and Intractability," *NY, NY: W.H. Freeman and Co.*, 1979.

[33] M. Gullickson, C. Eichholz, A. Chervenak and E. Zegura, "Using Experience to Guide Web Server Selection," *Multimedia Computing and Networking*, January 1999.

[34] A. Guyton and M. Schwartz, "Locating Nearby Copies of Replicated Internet Servers," *in Proc. of the ACM SIGCOMM'95 Conf.*, Cambridge, MA, pp. 288-298, Aug. 1995.

[35] J. Heidemann and V. Visweswaraiah, "Automatic Selection of Nearby Web Servers," *in Proc. of the 1998 SIGMETRICS/Performance Workshop on Internet Server Performance*, June 1998.

[36] Inktomi Content Delivery Suite, available at: http://www.inktomi.com.

[37] S. Irani, "A Competitive Analysis of Paging," available at: http://www.ics.uci.edu/~irani/.

[38] S. Jamin, C. Jin, Y. Jin, D. Riaz, Y. Shavitt and L. Zhang, "On the Placement of Internet Instrumentation," *in Proc. of the IEEE INFOCOM'00 Conf.*, March 2000.

[39] S. Jamin, C. Jin, T. Kurc, D. Raz and Y. Shavitt, "Constrained Mirror Placement on the Internet," *in Proc. of the IEEE INFOCOM 2001 Conf.*, Alaska, USA.

[40] Y.K. Kwok, K. Karlapalem, I. Ahmad and N.M. Pun, "Design and Evaluation of Data Allocation Algorithms for Distributed Database Systems," *IEEE Journal on Sel. areas in Commun.(Special Issue on Distributed Multimedia Systems)*, Vol. 14, No. 7, pp. 1332-1348, Sept. 1996.

[41] B. Li, M. Golin, G. Italiano and X. Deng, "On the Optimal Placement of Web Proxies in the Internet," *in Proc. of the IEEE INFOCOM'00 Conf.*, March 2000.

[42] T.D.C. Little and D. Venkatesh, "Popularity-Based Assignment of Movies to Storage Devices in a Video-on-Demand System," *ACM/Springer Multimedia Systems*, 1994.

[43] T. Loukopoulos and I. Ahmad, "Static and Adaptive Data Replication Algorithms for Fast Information Access in Large Distributed systems," i*n Proc. of the 20th IEEE Int. Conf. on Distributed Computing Systems*, Taipei, Taiwan, 2000.

[44] S. Mahmoud and J. Riordon, "Optimal Allocation of Resources in Distributed Information Networks," *ACM Trans. Database Systems*, 1(1), March 1976, pp. 66-78.

[45] P. Mirchandani and R. Francis, "Discrete Location Theory," *John Wiley and Sons*, 1990.

[46] P. Mockapetris, "Domain Names - Implementation and Specification," RFC 1035, ISI, Nov. 1987.

[47] K. Moore, J. Cox and S. Green, "Sonar - A Network Proximity Service," Internet Draft, Network Working Group, draft-moore-sonar-01.txt, Feb. 1996.

[48] J. Morris, M. Satyanarayanan, M. Conner, J. Howard, D. Rosenthal and F. Smith, "Andrew: A Distributed Personal Computing Environment," *Communications of the ACM*, Vol. 29, No.3, Mar. 1986, pp. 184-201.

[49] B. Narendran, S. Rangarajan and S. Yajnik, "Data Distribution Algorithms for Load Balanced Fault-Tolerant Web Access," *in Proc. of the 16th Symposium on Reliable Distributed Systems (SRDS '97)*, Oct. 22-24, 1997 Durham, NC.

[50] Probe Research Inc. available at: http://www.proberesearch.com.

[51] L. Qiu, V. Padmanabhan and G. Voelker, "On the Placement of Web Server Replicas," *in Proc. of the IEEE INFOCOM'01 Conf.*, Anchorage, AK, USA April 2001.

[52] M. Rabinovich, I. Rabinovich, R. Rajaraman and A. Aggarwal, "A Dynamic Object Replication and Migration Protocol for an Internet Hosting Service," *in Proc. of the 19th IEEE Int. Conf. on Distributed Computing Systems (ICDCS'99)*, May 1999.

[53] M. Rabinovich, "Issues in Web Content Replication," *in Data Engineering Bulletin*, Invited Paper, Vol.21 No.4, Dec. 1998.

[54] M. Sayal, Y. Breitbart, P. Scheuermann and R. Vingralek, "Selection Algorithms for Replicated Web Servers," *in Proc. of the 1998 Workshop on Internet Server Performance*, June 1998.

[55] A. Shaikh, R. Tewari and M. Agrawal, "On the Effectiveness of DNS-based Server Selection," *in Proc. of the IEEE INFOCOM'01 Conf.*, Alaska, USA, Apr. 2001.

[56] W. Tang, F. Du, M. Mutka, L. Ni and A.-H. Esfahanian, "Supporting Global Replicated Services by a Routing-Metric-Aware DNS," *in Proc. of the 2nd Int. Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems (WECWIS 2000)*, June, 2000.

[57] R. Tewari and N. Adam, "Distributed File Allocation with Consistency Constraints," *in Proc. of the 12th IEEE International Conference on Distributed Computing Systems (ICDCS'92)*, June 1992.

[58] A. Vigneron, L. Gao, M. Golin, G. Italiano and B. Li, "An Algorithm for Finding a k-median in a Directed Tree," *in Information Processing Letters*, 74 (2000), pp. 81-88.

[59] C. Yoshikawa, B. Chun, P. Eastham, A. Vahdat, T. Anderson and D. Culler, "Using Smart Clients to Build Scalable Services," *in Proc. of the 1997 USENIX Annual Technical Conference*, Jan. 6-10, 1997, Anaheim, CA, pp. 105-117, USENIX, January 1997.

[60] C. E. Wills and H. Shang, "The Contribution of DNS Lookup Costs to Web Object Retrieval," Technical Report WPI-CS-TR-00-12, Computer Science Department, Worcester Polytechnic Institute, July 2000.

[61] O. Wolfson, S. Jajodia and Y. Huang, "An Adaptive Data Replication Algorithm," *ACM Transactions on Database Systems*, Vol. 22(4), June 1997, pp. 255-314.

[62] World-Wide Web Consortium, "The HTTP Distribution and Replication Protocol," available at: http://www.w3.org/TR/NOTE-drp.