A Quorum Based Content Delivery Architecture

Michael P. Kapralos and John A. Chandy,
University of Connecticut
Department of Electrical and Computer Engineering
(michael.kapralos,john.chandy)@uconn.edu
Storrs, Connecticut, 06269-2157

Abstract

In this ever-growing world, the amount of information obtained by the human race is increasing at an exponential rate. However, along with this progress comes the need to develop the infrastructure to make this information accessible. Since a vast number of people are interested in accessing this information from a relatively small number of servers, an efficient algorithm must be developed and implemented to resolve this dilemma. It would be ideal that every user would be able to access only one server to get all the information. However, this is impractical due to storage limitations. It would also be inefficient because the data would have to be replicated and stored throughout multiple servers whenever a change was made to any or all of this information. This also leads the problem of mutual exclusion. If a server was updating the information, no information would be available to any other client on any other server. To make this information readily available, a new quorum-based scheme for managing this replicated data is developed and described in this paper. This quorum-based scheme will divide the servers into groups containing redundant data, and ensuring the clients that they can access information quickly without compromising network quality, also limiting the number of servers which must be read as well as written.

Keywords:
Content Delivery Networks, Force-Directed Placement, Quorums

1 Introduction

Information creation and consumption has exploded in the past decade and studies have suggested that the amount of information stored digitally will continue to increase for the foreseeable future [1]. It is estimated that the amount of information created per year has doubled from 2-3 exabytes in 1999 to over 5 exabytes in 2003, and that trend is expected to continue. This has been apparent in the recent growth of the Internet with the availability of large data archives on the web. For example, satellite imagery, genome sets, sports statistics, classic literature, maps, music and other data are all readily available at the click of a mouse.

The accessibility of all this information is often hindered by the location of that data at a single server. The single server can prove to be a bottleneck in terms of performance as well as prove to be a single point of failure and thus render the information completely inaccessible. The development of content delivery networks (CDN) is a response to this problem. In a CDN the content is judiciously replicated at several nodes across the Internet, thus enabling scalability as well as reliability. Companies such as Akamai, Digital Island, Speedera and others have established servers strategically placed at Internet access points across the Internet to provide the CDN replication.

For the most part, these CDNs are limited to read-only files, and are thus limited to applications that must serve multiple reads and very few writes. Synchronization is also not a key requirement as well.

For example, it is not particularly troublesome if a user gets an old page from a CDN cache even though the page may have been updated on a root server a few minutes before. For web serving, a read-only environment and loose synchronization are appropriate assumptions. However, in other applications such as mission critical data warehousing or transaction processing, these assumptions are not valid. In this paper, we present an architecture for mission critical content delivery networks based on the notion of quorums. We are particularly focused on the selection of nodes to serve as data servers in the presence of thousands of clients.

We first give the background and related work in the area of distributed storage and quorums, then describe our content delivery architecture, and then give results on various simulations of the system.

2 Background

Much work has been done in the area of content delivery-type networks particularly in the area of peer-to-peer networks. The OceanStore [2] storage architecture introduced many ideas in the areas of networks with large numbers of clients within the system. Securing data and ensuring the total reliability and accessibility of data, regardless of a large numbers of users leaving the system, due to Denials of Service attacks or other malicious and non-malicious changes in the network infrastructure. The CAN system[3] describes a similar methodology for fault tolerant decentralized networking system to OceanStore. While also similarly fault tolerant, they added a more robust and scalable identification mechanism. These structured peer-to-peer storage systems are focused on the routing of packets but do not address the distribution of servers in a client-server model and how to replicate data on those servers.

As a mechanism for determining how to replicate data in a distributed storage system, the use of quorums is particularly appealing especially since their construction is well understood in the context of distributed databases. In the context of distributed databases, the particular problem is the resolution of mutual exclusion. If a change is made to a piece of information, they must change each copy of the information, or a system must be set up to check each copy of the data. In order to do this, data must not be accessed as the data is changed. By not allowing multiple users to use the same piece of information and having a system of checking time stamps, the usage of corrupt and outdated data is prevented. This is despite an occasionally high penalty cost with regards to the speed in reading and/or saving data. As a result of these limitations, models for systems of networks, like the content delivery network model discussed in this paper, have been created which balance these costs and benefits in a way which makes the system as a whole function more efficiently. Various quorum design methodologies have been proposed including those based on combinatorial theory [4] and difference sets [5, 6].

3 Quorum Based Content Delivery Architecture

The basis of the content delivery architecture is the quorum, whereby for each piece of data we define a set of servers that will store that data. This set is the *write* set. Likewise, we also define a set of servers that a client can contact to retrieve any piece of data. This is set is the *read* set. Obviously, in order to insure that all data is accessible, each read set must intersect with all possible write sets. The size of the write set is the degree of replication. The construction of these quorum sets is a well studied area, and in the design of our architecture, we use the quorum design methodology proposed by Lin et al [6]. However, since our content delivery network architecture is quite flexible, we are not restricted to that particular construction algorithm.

Lin, et al.[6] have described quorum based data replication schemes and outlined the basic design methodology for laying out systems of replicated servers within a highly scalable context. By proving the validity of using such a system, real implementations and simulations, such as the one implemented and illustrated within this paper, may be designed. The basic structures of quorums and coteries

are shown with extensive mathematical proof, giving a solid foundation for this simulation model. We first establish some important equations used by Lin et al in the quorum construction. The sizes of read and write quorums are established by the following 2 equations. Overall, the sizes are defined as follows:

$$l = \lceil \frac{N}{m} \rceil \tag{1}$$

$$k = \lceil \frac{N+1}{2m} \rceil \tag{2}$$

where the size of the read and write generation sets are of size l and and $m+k-1$, and N is defined as the number of copy sites for a particular data set. Each of these sets of servers are proven to contain all of the sites a client needs to check in order to possess all of the data required within the system, without making unnecessary reads. m is a constant that changes with each implementation. It adjusts the proportion of read and write quorums to maintain data integrity. This constant would usually be obtained experimentally or empirically given a particular problem, based on the read versus write frequency in a system. The servers in each quorum are laid out using a methodology called a difference set. This difference set has been proven to have the desirable properties within a quorum. The basic units for the difference sets for this model is called a (N, k, l) difference pair where N, k, and l are defined as above. Through this setup, the derivation of the quorums are quite simple to implement to simplify the ability to solve the larger problem.

Once the quorum has been constructed, the question becomes how to map particular servers in a network to nodes in a quorum set. Specifically, for each client, which read set does it belong to, and can the mapping of the read set to actual servers be optimized for performance. Performance is optimized by selecting servers such that the network distance between client and server is minimized. Since the problem is NP-complete, a closed-form solution is not available and more heuristic approaches are required.

The approach we use is based on force directed placement techniques using spring models. Force directed techniques have been used in a variety of applications including graph drawing [7, 8, 9] and VLSI placement [10, 11, 12] Since there is a large body of work in the areas of forced direction and graph drawing, and much is known about their properties, this seems like a logical model to explore for the solution to content delivery networks. The goal in graph drawing is to draw aesthetically pleasing graphs. While aesthetics are not applicable in the application of content delivery networks, graph drawing does motivate the idea of creating virtual springs and solving for a steady state on these large systems of springs.

The initial step in modeling the content delivery network is to map the network onto a graph where each of the vertices of the graph are clients and servers. The optimal result for the cost function associated with the force direction would be the most optimal quorum set up in a methods defined in Lin, Chiu, and Cho [6]. Our forced placement model contains 2 basic forces on the server. An attractive force is defined as the ratio of bandwidth of the server and the number of clients using that server. This spring would go to each individual client, but the sum of these springs would be the bandwidth of the server. A repulsive force is also placed between the clients and servers within the system. The coefficient of this spring is the ”distance” between the server and the individual clients. Generally it can be expressed that

$$Force \propto \frac{bw/users}{d} \tag{3}$$

.

The forces in the model are defined as sets of linear springs between each client server reaction. An attractive spring, one which is defined as a spring with a negative coefficient relating to the proportion of bandwidth to the number of users. A repulsive spring attached between a server and client would be attached with the coefficient being the ”distance” between the client and server. This distance is not necessarily a linear distance or a space like the d-torus used in Ratnasamy et. al. [3], per se, rather, an effective distance relating the number of hops, and therefore time, used in a connection between the server and the client. Ideally, a server would have an in-

finitely negative number where the number of users is zero and the bandwidth is high. The "least positive" server would be therefore chosen as the ideal server. Also, the attractive forces are only active when the client has chosen a particular server as belonging to its ideal quorum.

A few basic assumptions had to be made for this model. First, information about the locations of each client and server should be known, whether they are empirical or actual locations at the time of a client's entry into the content delivery network. Also, the key assumption made is that the information about the servers are not changing at such a rate that the calculations would take longer than the period within which the solution would be valid. If this were the case, more probabilistic models would be more appropriate.

Although the basic layout of the quorum is not strictly defined in Lin, Chiu, and Cho [6], one has been chosen for this particular problem. In this type network, a lower rate of writing compared to the number of reads is to be assumed. For example, the popular news site might be written at most a few dozen times a day, while it would see several million visitors a day. Since these proportions are differing by several orders of magnitude, a larger write penalty for a given write quorum can be established while keeping the size of the read quorum relatively small. This data also needs to be stored in more than one location so that high access speeds can be maintained regardless of location. Exact duplications of all data within each quorum is not an acceptable solution, because either a simultaneous and catastrophic failure might cause the loss of all the copies of a set of data.

Algorithm 1 Spring Algorithm

1: Place vertices in the locations of the server
2: Place spring from each server to each client
3: Select initial server based on the distance from the client
4: **for** $i = 0$ to X **do**
5: Find the server with the lowest force total for the given client
6: Remove force from client on old server
7: Add force to the new server
8: **end for**
9: Add additional clients as they join the system, and repeat

4 Results

We performed simulations to evaluate the model and quorum based content delivery network. A relatively small set of clients and servers have been chosen for ease in simulating the results, but our model is not limited in any way. 500 clients have been selected and 100 servers have been used with a read quorum of size 9, which implicitly leads to the write quorum being of size 14. Although these particular numbers are not in line with any particular model, they are reasonable within any number of content delivery networks' contexts.

To do our tests, 20 different simulations were run. In the first 10, the bandwidth of each server varied widely, from 1 megabit per second to 1 gigabit per second, using a uniform distribution. The second set of 10, also uniformly distributed, had 3 possibilities 200, 250, and 400 megabits per second of bandwidth available. The locations of both the servers and the clients were set up in a 360 by 360 grid.

Method	Percent Servers Used	Bandwidth
Distance	84.7	1521
Random	90.1	1232
Our Method	85.7	1600

Table 1: Quorum Based CDN Simulation Results

As can be seen from Table 1, distance alone is a poor indicator, for both available bandwidth for the quorum and the percentage of servers used. While the random decision process works better to get a higher number of used servers than our process, it does so at the cost of effectiveness, losing out significantly in the area of total bandwidth.

Our results shows that by using a simple spring model, a much more efficient usage of servers can be laid out. For this reason, our model for content delivery networks has been show to be a scalable, efficient

one. We expect that more specific information, particularly information with population-centric client locations the results would be give similar results and the other 2 methods might produce even worse results.

5 Future Work

Ideally, the next work on the subject would include validation of simulation results with tests from actual networks, severs, and clients. By doing this, weakness in the model, if any, could be shown, and constants could be manipulated in such a way to further optimize performance. These models could then be implemented and ported to other types of systems, not just content delivery networks.

In addition, we see opportunities to use a quorum based architecture to setup networks of wireless sensors. These networks face additional constraints that many other network systems do not. The largest of these revolves around finite battery life. As a result of this, computations must be limited so that the longevity of the sensor and the usefulness of the network systems is maximized. Since this power constraint exists, this content delivery network model must be modified and analyzed to examine the exact penalties paid for not selecting the most optimal quorum at each client's arrival, due to the lack of the ability to recompute the spring forces each time because of limited benefit at this level.

Finally, in areas where files are larger than a single source could contain, connections must be made with multiple servers for pieces of the same file. For this reason, TCP re-direction seems like an elegant solution to large groups of replicated servers across multiple locations. Implementation of solutions to this problem would provide a valuable service to the Internet community given the relative inefficiencies involved in relaying information between servers before forwarding it to the client.

6 Conclusion

In the end, we were able to successfully design a full content delivery network capable of handling an indefinite amount of clients and servers, and then implement the usage of quorums on the network to indicate the model's efficiency. We also showed that our algorithm is efficient, and robust, not computationally burdensome. This modeling information may be used in future implementation and testing to do such activities as indicated where new servers should be placed for optimal efficiency based on client access location or which servers should be given more bandwidth to increase their usage. Also, further application may be derived from our work, such as newer algorithms for making quorums. We are confident that research we have done, and the new information we have produced by relating quorums and coteries to content delivery networks will be useful in producing more algorithms in the future to further optimize networks.

7 Acknowledgments

We would like to thank Cory Liu for his invaluable discussions and help in the creation and development of the model for this paper.

References

[1] P. Lyman and H. R. Varian, "How Much Information?." http://www.sims.berkeley.edu-/research/projects/how-much-info-2003, 2003.

[2] J. Kubiatowicz, D. Bindel, Y. Chen, P. Eaton, D. Geels, R. Gummadi, S. Rhea, H. Weatherspoon, W. Weimer, C. Wells, and B. Zhao, "OceanStore: An architecture for global-scale persistent storage," in *Proceedings of the International Conference on Architectural Support for Programming Languages and Operating Systems*, Nov. 2000.

[3] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content-addressable network," in *Proceedings of SIGCOMM*, pp. 161–172, Aug. 2001.

[4] M. Maekawa, "A $\sqrt{N}$ algorithm for mutual exclusion in decentralized systems," *ACM Trans. Computing Systems*, pp. 145–159, May 1985.

[5] W.-S. Luk and T.-T. Wong, "Two new quorum based algorithms for distributed mutual exclusion," in *Proceedings of the International Conference on Distributed Computing Systems*, May 1997.

[6] C.-M. Lin, G.-M. Chiu, and C.-H. Cho, "A new quorum-based scheme for managing replicated data in distributed systems," *IEEE Transactions on Computers*, vol. 51, pp. 1442–1447, Dec. 2002.

[7] P. Eades, "A heuristic for graph drawing," *Congressus Nutnerantiunt*, vol. 42, pp. 149–160, 1984.

[8] T. M. J. Fruchterman and E. M. Reingold, "Graph drawing by force-directed placement," *Software Practice and Experience*, vol. 21, pp. 1129–1165, Nov. 1991.

[9] P. Gajer and S. G. Kobourov, "GRIP: Graph drawing with intelligent placement," *Journal of Graph Algorithms and Applications*, vol. 6, no. 3, pp. 203–224, 2002.

[10] C. J. Fisk, D. L. Caskey, and L. E. West, "ACCEL: Automated circuit card etching layout," *Proceedings of the IEEE*, vol. 55, pp. 1971–1982, Nov. 1967.

[11] N. R. Quinn and M. A. Breuer, "A force directed component placement procedure for printed circuit boards," *IEEE Transactions on Circuits and Systems*, pp. 377–388, June 1979.

[12] K. J. Antreich, F. M. Johannes, and F. H. Kirsch, "A new approach for solving the placement problem using force models," in *Proceedings of the International Symposium on Circuits and Systems*, pp. 481–486, 1982.