Replication over the Internet

Computer Science and Engineering
University of Texas at Arlington
Team 3B

May 2nd, 2002

Abstract

The immense success of the World Wide Web has given to rise to serious problems due to the inability of the existing resources to cope up with the increasing demands. The continuous increase in the number of applications that use the web's infrastructure for various purposes has further intensified the problem. In addition to these problems a dramatic increase in the number of users, coupled with multimedia based information services, real-time audio/video transmissions and the emergence of network commerce are contributing to an incredible demand for bandwidth. This enormous load on web servers and the inability of caching mechanisms to keep pace with the heavily increasing demand for web documents has resulted in high latencies, low throughput and huge amounts of network traffic. As of today, the requirement for a better web is to deliver the information in a speedy, efficient and convenient manner. In order to fulfill these requirements the technique of replicating documents within the organization or geographical region was considered. This approach provides faster access to documents, while reducing the network traffic and web server load. This paper examines the replication techniques for different architectures and gives a comparison with caching in different architectures.

1. Introduction

As commercial interest in the Internet continues to grow, the issues of scalability and performance become increasingly important. In an effort to alleviate this growth induced pain the networking industry has come up with various ways. A common way to do this is to upgrade the loaded resource: a faster server, a bigger switch, re-engineering the network. However this approach is not always economically feasible and more importantly, it also fails to consider the number of users. Consequently, caching and replication, being the primary tools that address these issues, have become the focus of attention of the industry and academic research communities. In the Web, much attention has been paid towards caching. Recently, it has been recognized that caching alone is not enough. In particular, replication techniques for updating the clients are needed. Also *many objects are not cacheable* but replicable. These include dynamic objects with read-only access and personalized objects. In addition, many objects that are updated as a result of accesses are still replicable because updates commute. Now lets understand the basic difference between caching and replication.

Caching is storing an object at a place that sees the object anyway in the course of request processing. Examples include browser cache, proxy cache, and server main memory cache. ***Replication*** is storing an object at a place that would not otherwise see the object. Web site mirroring provides an example of replication. Alternatively, one can say that the difference between a cache and a replica is that a cache routinely sees both hit and miss requests, while a replica normally sees only hits except when a request arrives for an object that has been deleted. In other words, requests flow to the cache regardless of the cache contents, while requests arrive at a replicated server only if that server is believed to have a replica of the requested object. So, the process of replication is to copy the cache content and push it on to one or more replica servers across the network. Replication is required to distribute objects among the servers to maintain the freshness of content across servers, which results in reduced upstream network traffic. Typically the same content is pushed across several machines making it more efficient to use multicast. Replication is critical in global operations, where cost of international traffic is high and ways have to be found to mirror data without using too much bandwidth. On the other hand a web cache is a dedicated computer system that will monitor the object requests and stores objects as it retrieves them from the server. On subsequent requests the cache will deliver objects from its storage than passing the request to the origin server. Every web object changes over time and therefore has a useful life or "freshness". If the freshness of an object expires it is the responsibility of the Web cache to get the new version of the object.

With organizations supporting diverse hardware and software applications in distributed environments, it becomes necessary to store data redundantly. Moreover, different applications have different needs for autonomy and data consistency. Replication is a solution for a distributed data environment when you need to:

- Copy and distribute data to one or more sites.
- Distribute copies of data on a scheduled basis.
- Distribute data changes to other servers.
- Allow multiple users and sites to make changes then merge the data modifications together, potentially identifying and resolving conflicts.
- Build data applications that need to be used in online and offline environments.
- Build web applications where users can browse large volumes of data.

The paper is organized as follows: Section 2 discusses the different replication techniques used currently. Section 3 examines the different architectures in which replication is implemented or researched. Section 4 gives our design and implementation of replication on Internet with the simulation results. Section 5 gives a comparison of replication against caching. Finally Section 7 concludes by discussing the effects of replication on the web.

2. Replication techniques

In this section, we present four replication techniques that have been proposed in the literature in the context of distributed systems.Redundancy is usually introduced by the replication of components, or services. Replicating a service also requires that each replica of the service should have a consistent state. This consistency is ensured by a specific replication protocol. Some replication protocols can be found in [1].

There are essentially two main classes of replication techniques:
1) Active Replication.
2) Passive Replication

2.1. Active Replication

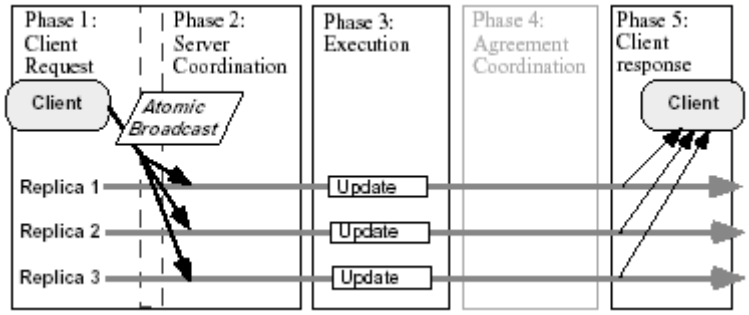

Fig 2.1 Active Replication

It is also known as the ***state machine approach***. Active Replication is illustrated in Fig 2.1. It is a *non-centralized replication technique*. The main idea of this is to ensure that all replicas receive and process the same order/sequence of client requests. Consistency is guaranteed here by making the assumption that, when provided with the same input in the same order, replicas will produce the same output. So the servers have to process the requests in a *deterministic* way. The meaning of *Determinism* is that the result of an operation depends only on a replica's initial state and the sequence of operations it has already performed. Clients do not contact any one particular server, but address servers as a group. Client requests should be then propagated to servers using an Atomic Broadcast, so that servers receive the same input in the same order. The main advantages of active replication are its simplicity and failure transparency.

2.2. Passive Replication

It is also known as ***primary backup*** replication. Passive Replication is illustrated in Fig 2.2. In this passive replication technique, any one of the replicas, called the *primary*, plays an important role. It receives the requests from the clients and returns responses. So initially the clients send their requests to this primary, which it then executes the requests and sends update messages to the backups. The backups do not execute the invocation, but apply the changes produced by the invocation execution at the primary. So this is

quite different from active replication, no determinism constraint is necessary on the execution of invocations in passive replication.

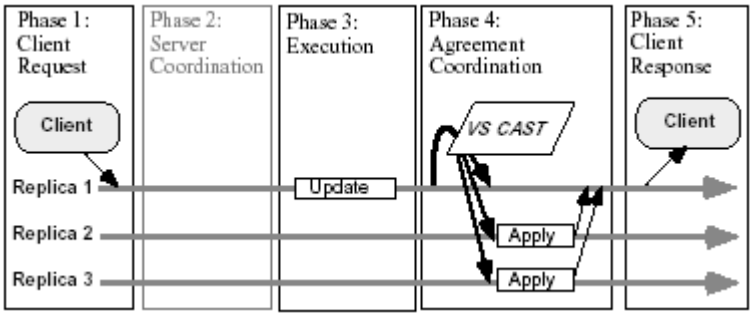

Fig 2.2. Passive Replication

Communication between the primary and the backups has to guarantee that updates are received and then processed in the same order, which is the case if primary backup communication is based on FIFO channels. However, FIFO channels are not enough to ensure correct execution in case of failure of the primary. For example, consider that the primary fails before all backups receive the updates for a certain request, and another replica takes over as a new primary. Some mechanism has to ensure that updates sent by the new primary will be "properly" ordered with regard to the updates sent by the faulty primary. VSCAST is one mechanism that can be used to implement this primary backup replication technique.

2.3. Semi-Active Replication

Semi-active replication is an in-between solution between active and passive replication. Semi-active replication does not require that replicas process service invocation in a deterministic manner as was done in active replication. The protocol was originally proposed in a synchronous model. The main difference between semi-active replication and active replication is that each time replicas have to make a non-deterministic decision, a process, called the *leader* makes the choice and sends it to the *followers*.

2.4. Semi-Passive Replication

Semi-passive replication is a just a small variation of passive replication, which can be implemented, in the asynchronous model. The main advantage over passive replication is to allow for aggressive time-outs values and suspecting crashed processes without incurring too high a cost for incorrect failure suspicions. In semi-passive replication the Server Coordination (phase 2) and the Agreement Coordination (phase 4) are part of one single coordination protocol called *Consensus with Deferred Initial Value*s.

3. Replication in different Architectures

3.1. Internet

As traffic increases in the web, there is a gradual degradation in its performance. The web is not designed to be suited for a large number of users. Today's Internet requires data to be transmitted in a speedy, efficient and convenient manner. So in order to fulfill this requirement replication is used. The main aim of replication is to increase document availability and latency. Also bandwidth balancing and backward compatibility may also be integrated in a replication scheme.

The idea of replication is simple. The basic scheme behind replication is to keep multiple copies of the same resource or type at various servers. So by this method a client can contact any of the nearby replica servers, preferably the one that is closest. This helps in improving latency and congestion in Internet. Different models/schemes have been proposed for replication in Internet. Even schemes have been proposed to integrate caching with replication. There are many issues, which we need to consider while designing the architecture. Some are as follows:

- Consistency of the replicas in the different servers.
- Redirection of a request to a replica.
- Selecting the replicas

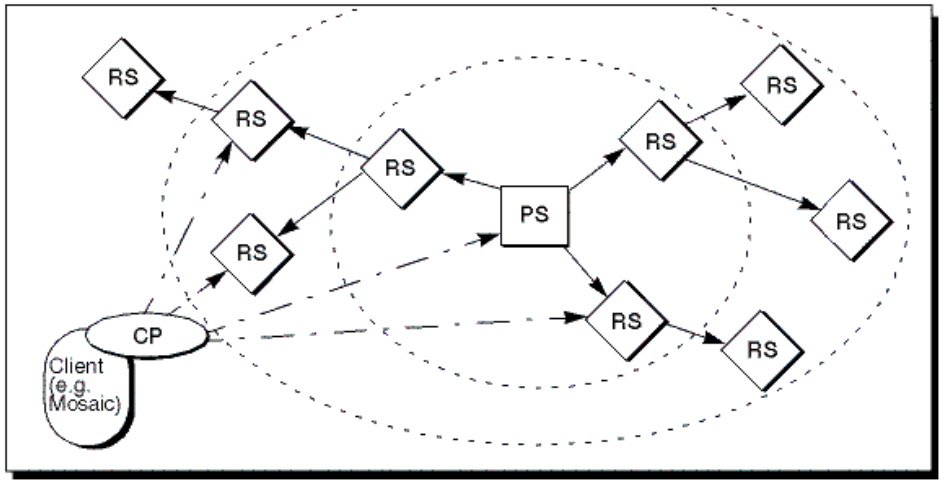

Fig 3.1. Overview of the CgR scheme

One scheme for replication is mentioned in [2]. This scheme, which is called ***Caching goes Replication*** (CgR), has been designed to meet the criteria for a better and more efficient web. This is shown in Fig 3.1. There is one ***Primary Server*** **(PS)**, which stands at the center of a hierarchically structured group of ***Replicated Servers*** **(RS)**, which serve the PS's replicated namespace. The RSs are updated automatically by the PS. No RS is exclusively dedicated to one PS, but may replicate URL namespaces of many primary servers. Users access the system via ***Client-side Proxies (CP),*** which know the concept of CgR and act accordingly. RSs are used for obtaining any information as soon as they are

in any way better accessible than the PS; CPs may switch RS anytime for load balancing purposes or for fault tolerance.

3.2. Mobile Platforms

Replication is an essential utility in ***mobile environments***. The architecture of a typical cellular network is shown in Fig 3.2.

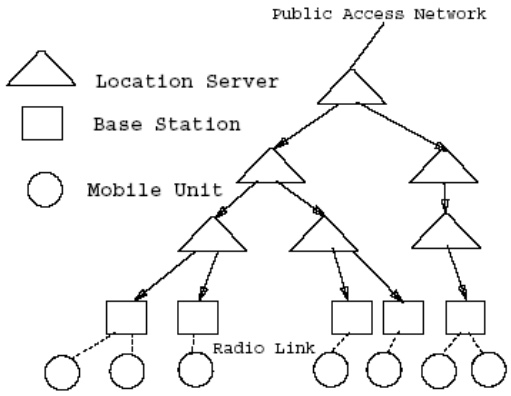

Fig 3.2 Architecture of a typical cellular network

Mobile units not connected to fixed networks are left with their own data resources, which may or may not contain the necessary data. Mobile users access data that are also accessed by other users or itself on different locations and devices respectively. Because of rare, sporadic and weak connections, a local copy has to be maintained on the mobile device to avoid the establishment of an expensive connection and the remote accessing. Such a copy increases availability, decreases answer time and also improves fault tolerance. Replication is one of the key problems in mobile information systems and is extensively being researched on. Replication seeks to provide high availability on mobile sites especially during disconnected phases, consistency and low communication costs.

There are many different replication strategies in mobile environments. Enumerating all of the schemes is beyond the scope of this paper. Among them some are as follows with special emphasis on the third strategy.

- The paper [3] seeks to implement replication using ***existing replication*** techniques.
- The paper [*] sought out a ***mixed mode replication***, where pessimistically controlled replicas coexist with optimistically controlled ones. Operations are grouped into three sets of operations which 1.cannot invalidate others. 2. Be invalidated by others. 3. The set of all operations. This approach improved the consistency of mobile copies.
- In the paper [4] the author proposes many strategies.

Let ***s*** be the *server* and ***c*** be a *client* and ***x*** be the *item*, which is written by the server and read by the client. This basic scenario in Fig 3.4 is a simple hierarchical network with L levels of location servers. Here a simple scenario of a single client and a single server within their home location servers is considered. It is assumed that the server has a copy of the data and writes, whereas the client only reads the copy of data.

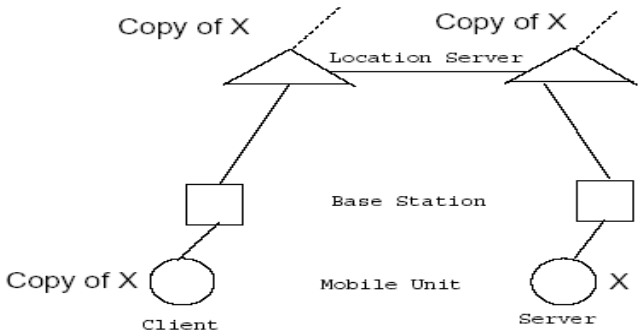

Fig 3.4 Possible places of a Replicated Copy

There are many schemes proposed for placing a replicated copy. The main schemes are:

- The Server replicates the copy of the data at the mobile client. On each write, the server needs to write to the copy on the mobile client. Writing requires locating the mobile client. Reading is from a local copy on the mobile client.
- The replicated copy resides at the location server of the client. Thus the client reads from its own location server. Here, read and writes are on static copies only. However, the copy is closer to the reader than the writer.
- The server S has a copy of the data at its home location server Ls. The client reads from Ls. Thus reading and writing is on static remote copies.

3.3. Large-scale Distributed File systems.

The ever-increasing need for data sharing in large scale distributed systems degrades the performance of file servers and networks. One such scheme to alleviate such problems is ***FROLIC*** (***File Replication over Large Interconnected Clusters***), which is a scheme for cluster based file replication in large-scale distributed systems. A ***cluster*** is a group of workstations and one or more file servers on a local area network. A ***Large-scale distributed systems*** have tens or even thousands of clusters connected by a backbone network. Fig 3.5 shows a typical cluster large-scale distributed system. *Traditional client based caching techniques are not scalable in such environments.*

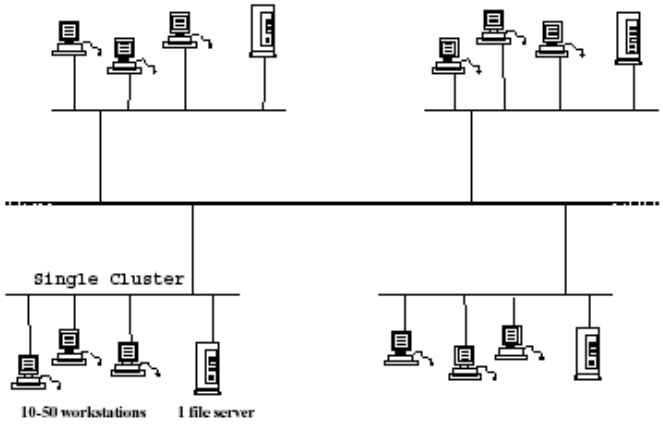

Fig 3.5 A Cluster large scale Distributed System

The replication model is shown in Fig 3.6. ***FROLIC*** approach involves dynamically, creating and maintaining replicas of files on the file servers within the clusters that access the files. By this replication across the servers, there is a change from n to 1 client-server relationship to an n to k to 1 client server relationship. So by maintaining replicas within clusters where they are accessed, there is also a reduction in *access latency* and *file access* delays although a local server will need to obtain a copy of the file before replying to the client.

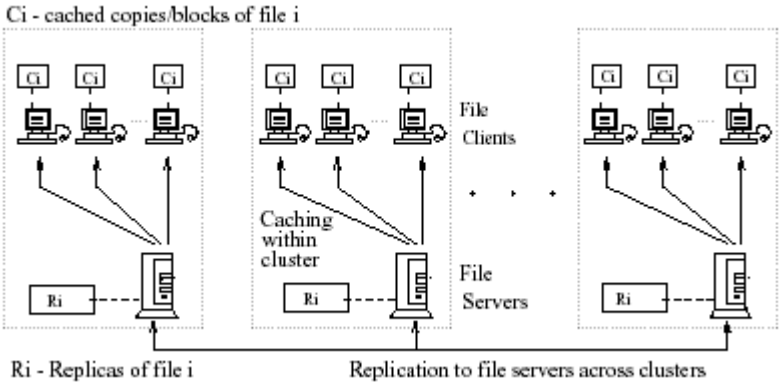

Fig 3.6 Dynamic cluster based file replication

3.4. Streaming Media

Streaming means the way that playable media audio or video is delivered on a typical network. Replication is ideally suited for ***on-demand streaming*** and for ***pre-recorded live events***. Using these replication techniques, one or more copies of a single streaming media resource or a whole file, containing multiple streaming media resources, can be maintained on one or more different servers, called 'replica origin servers'. A client can contact one or more '***replica origin servers'***, as well as with '***master origin servers'***. In the absence of replica servers the client interact directly with the origin server, as is the normal case. The Fig 3.7 shows a basic model of replication of video content.

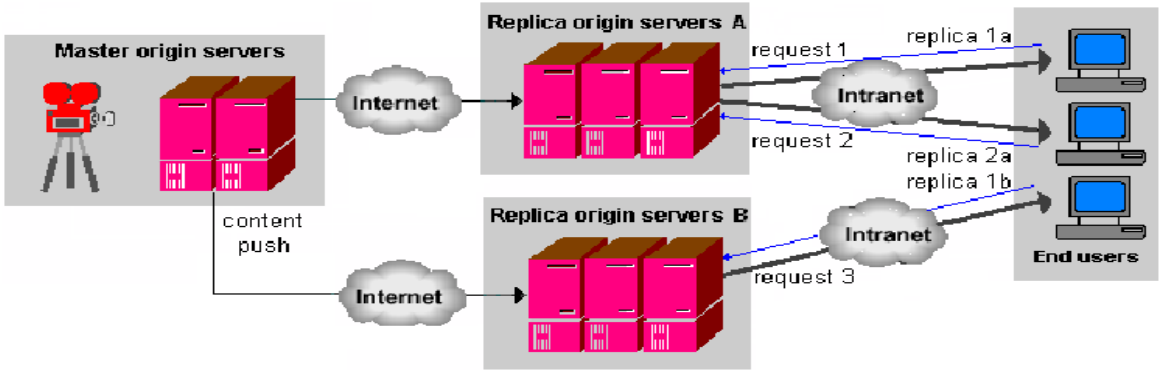

Fig 3.7 Replication of Video Content

In this type of model the client to replica communication is done by ***hyperlink URLs*** embedded in web pages, which point to the respective individual replica origin server the client wants to use. There is another method used to connect clients with a replica origin server is to use ***HTTP redirection***. Also clients can use the ***Domain Name Service,*** which provides a more advanced and better client to replica communication. When a client resolves the name of an origin server, the DNS server sorts the available IP addresses of the replica origin servers starting with the most optimal replica and ending with the least optimal replica. Content delivery networks such as *Akamai*, *Digital Island* and *Mirror Images* are using this approach. Also another problem is to synchronization of the data between the master and replica servers. A commonly used method is using "*batch driven replication protocols*" such as '*RSYNC*', '*FTP*' or "*RDIST*'. The replica origin server to be updated periodically initiates communication with the master origin server. The communication is established at intervals based upon queued transactions that are scheduled for deferred processing. Once communication is established, data sets are copied to the initiating replica origin server. Another method is using "**demand driven replication**", in which the replica origin server will acquire the content as needed due to client demand. When a client requests a resource that is not in the data set of the replica origin server, an attempt is made to satisfy the request by acquiring the resource from the master origin server, and returning it to the requesting client.

4. Project Design and Implementation

4.1. Aim:

The best effort service provided by IP network has necessitated development of middleware tools to overcome insufficient bandwidth and high latency in provisioning QoS on the Internet. Several techniques like Caching and Replication have been developed to meet the requirements of faster and efficient Internet. The aim of our project is to design and implement a simulation environment to show the working of Replication over the Internet. There are certain assumptions made:

- All the data in *Origin Server [OS]* are replicated to the *Replicated Servers [RS]*.
- Replicated servers are already set up based on user access patterns.
- The DNS has the IP addresses of all the RS and also the OS.
- Updates made to the file are propagated by the OS to the replicated server.
- File size is big (in the order of MB) and are seldom updated at the origin server

4.2. Components of the Simulation Environment :

The fig 4.1 shows a basic design of the proposed simulated environment. The components involved are:

- *Origin Server [OS]* is the main server as the name itself suggests. The Origin propagates any updates to the Replicated Servers as and when any file is changed.

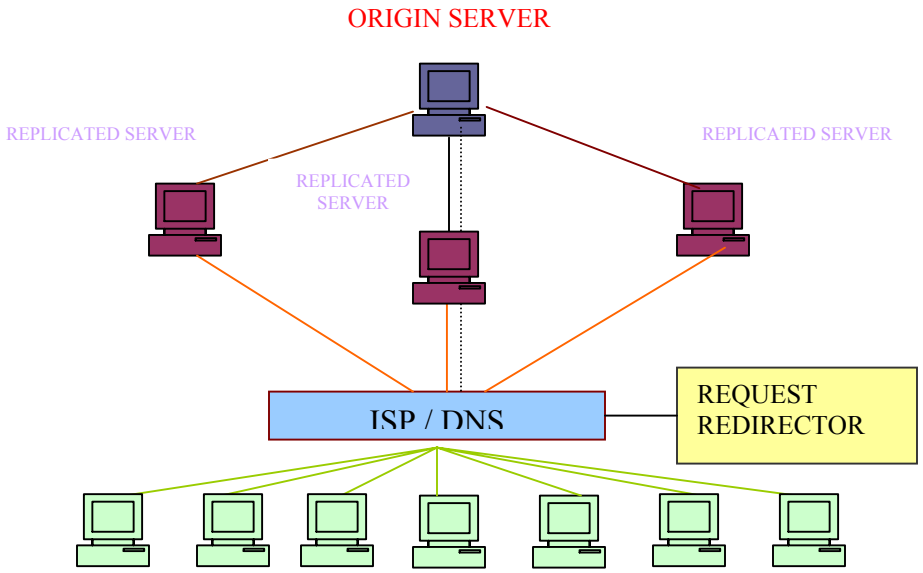

Fig 4.1. ARCHITECTURE

- *Replicates Server [RS]* is a mirror site dedicated to a particular origin server i.e. all the data from the OS is replicated onto the RS
- *ISP / DNS:* The Internet Service provider [ISP] provides connection to the Internet for the various clients. The Domain Name Service [DNS] maintains a list of OS's IP address and also all the RS's IP addresses .
- *Client Request Redirector:* This is primarily the module implemented at the DNS to perform *Dynamic Replica Server Selection* when a client request arrives at the DNS
- *Clients* The user requesting for files

4.3. Data structures Maintained:

- *Table of IP Addresses of Replica servers* is maintained at the **Origin Server**. This helps the origin server to send the updates to any file by multicasting the updates to all of them.
- *Table of IP addresses of Web Servers (both origin servers and replica servers)* is maintained at the **DNS**.

4.4. Steps in Simulation:

- The Domain Name Server is started at a specified port.
- The Origin Servers registers itself to the DNS
- The Replica Servers are started and they register themselves to the Origin Server and the DNS
- The Clients are started and a request for a file is sent to through the DNS
- If the DNS finds that there is a replica found for that file, the Request redirector is invoked.
- The Request Redirector module of the DNS makes a decision as to which replication server should be contacted for the file.

4.5. Request Redirector Module :

The request redirector uses the policy of Dynamic Server Selection to redirect the client requests to any one of the replica servers. The purpose of doing this being, to distribute the server load and the network load when large files are requested. The steps implemented at the redirector are

- Based on the table of addresses maintained at the DNS, the redirector continuously performs '*ping*' requests to the origin server and the corresponding replica servers. Based on the Round-Trip times(RTT) of these ping requests, any congestion in any of the links is detected.
- An average of five ping measurements is taken and based on this a *rank* is assigned to the origin server and each of the replica servers .(Lower the RTT higher the rank and vice versa) . These servers are arranged in the decreasing order of their ranks.
- Then for each client request the redirector follows a Round-Robin Fashion and chooses the replica servers.
- In the background, the redirector module does ping measurements. So the rank of the servers keeps changing dynamically based on the network load .

5. Comparison

Caching systems:

- Reduce network latency by bringing content closer to the content consumer.
- Essentially reactive wherein a data object is cached only when the client requests it.
- Meet traffic reduction goals by only getting content when requested.
- Consistency problems due to their reactive nature
- Reliability problems as they are normally placed at network entry points and a cache failure may sometimes bring the whole network down.

Replication systems:

- Exactly knows when an object changes and push the objects immediately.
- Ensures content freshness due to their reactive nature.
- Very high fault tolerance due to replication of data, which ensures that even if a web server goes down requests can be redirected to another origin server.
- Knowledge of the persistent domain allows load balancing.
- Consume more disk space.
- Need efficient algorithms for load balancing.
- May increase network traffic if Multicast is not used judiciously.

6. Discussion and Conclusion

In the World Wide Web, information is stored in the form of multimedia documents. Traditionally, a web server has the documents and delivers a document in response to the requests made by clients. The tremendous growth of the Web has resulted in a huge amount of requests. The single server approach is not able to cope up with the increasing

demand, which leads to server overloading, network congestion and an increase in the document latency time. This paper outlines the need for replication of resources on the Internet to combat this huge growth in user population and bandwidth requirements. It describes deficiencies in caching scheme for document replication and shows approaches using replication for more widely distributed load across multiple servers.
Our project gives a brief idea of how Replication can be implemented over the Internet. The main QoS features considered were the document retrieval latency, data availability, network resource utilization, network congestion and scalability. Replication would definitely be a better choice for the above features if the data considered were to be big, and audio/video transmissions. The will be a slight overhead of maintaining a replicated server but will be a very good trade off for the Quality of Service provided to the user.

7. References:

1) R. Guerraoui and A. Schiper, *"Fault-tolerance: from replication techniques to group communication," IEEE Computer, vol. 30, pp. 68–74, Apr. 1997. 88*
2) Baentsch, L. Baum, G. Molter, S. Rothkugel, and P. Sturm, "*Enhancing the Web's Infrastructure: From Caching to Replication", IEEE Internet Computing, Vol. 1, No. 2, pp. 18-27, March-April 1997*
3) D.Barbara and H Garcica-Molina. *"Replicated data management in mobile environments: Anything new under the sun?" In IFIP conference on Applications in parallel and Distributed Computing, 1994*
4) B. R. Badrinath and T. Imielinski. "*Replication and mobility". In Proc Management of Replicated Data, pages 9--12, 1992.*
5) M. Wiesmann, F. Pedone, A. Schiper, B. Kemme and G. Alonso. "*Understanding replication in databases and distributed systems*". *In Proceedings of the 20th international conference on distributed computing systems (ICDCS)*
6) X. Defago, A. Schiper, and N. Sergent. *"Semi-passive replication". In Proceedings of the 17th IEEE Symposium on Reliable Distributed Systems (SRDS), pages 43--50, West Lafayette, IN, USA, Oct. 1998.*
7) H. Sandhu and S. *Zhou "Cluster-based file replication in large-scale distributed systems". In Proc SIGMETRICS, June 1992*
8) Michael Rainovich " *Issues in Web Content Replication". Technical Report AT &T labs.*
9) Andy Meyers, Peter Dinda and Hui Zhang, "*Performance Characteristics of Mirror Servers on the Internet" A DARPA sponsored project*.
10) Ellen W. Zegura, Mostafa H. Ammar, Zonming Fei and Samrat Banerjee, "*Application Layer Anycasting : A Server Selection Architecture and Use in a Replicated Web Service*
11) Robert L.Carter and Mark E. Crovella, " *Server Selection using Dynamic Path Characterization in Wide-Area Networks", IEEE Computing 1997.*
11) Hamesh Chawla and Riccardo Bettati, " *Replicating IP Services", Technical Report 97-008*