Introduction

The IT infrastructure is playing an increasingly important role in the success of a business. Market share, customer satisfaction and company image are all intertwined with the consistent availability of a company’s website. Network servers are now frequently used to host ERP, e-commerce and a myriad of other applications. The foundation of these sites – the e-business infrastructure – is expected to provide high performance, high availability, and secure and scalable solutions to support all applications at all times. However, the availability of these applications is often threatened by network overloads as well as server and application failures. Resource utilization is often out of balance, resulting in the lower-performance resources being overloaded with requests while the high-performance resources remain idle. Network load balancing is a widely adopted solution to performance and availability problems.

Also the number of users accessing the internet is increasing rapidly and it is not uncommon for popular sites to get hundred million hits a day. Replication of web servers is one way to deal with a large number of requests. However, to improve performance, effective load balancing schemes are needed. Network load balancing is the process of distributing service requests across a group of servers. It is a computer networking methodology to distribute workload across multiple computers or a computer cluster, network links, central processing units, disk drives, or other resources, to achieve optimal resource utilization, maximize throughput, minimize response time, and avoid overload.

In this project we are removing the problem of increased load by using the concept of clusters and load balancing schemes such as weighting, randomization, least connections, minimum misses and round robin DNS. After applying these all methods load get balanced among the servers of a cluster.

Scope

In the current scenario, the use of internet is increasing rapidly and therefore load is getting increased on the servers. Network Load Balancing servers (also called hosts) in a cluster communicate among themselves to provide key benefits, including:

- **Scalability.** Network Load Balancing scales the performance of a server-based program, such as a Web server, by distributing its client requests across multiple servers within the cluster. As traffic increases, additional
servers can be added to the cluster, with up to 32 servers possible in any one cluster.

- **High availability.** Network Load Balancing provides high availability by automatically detecting the failure of a server and repartitioning client traffic among the remaining servers within ten seconds, while providing users with continuous service.

A part from that we can also include some very common real life examples:

As we all know that Google is the world’s one of the most busy site. Millions of hits are encountered on the Google within seconds; a lot of traffic is produced under these conditions so some times when traffic crosses beyond a certain limit the site gets blocked. The users are not able to access the site at that time; this is due to the increasing load on the site. So in these kinds of conditions network load balancing plays its role by shifting the load among different ip addresses in the cluster so the user can access the site at any particular instance or in any condition.

**Problem Statement**

Many content-intensive applications have scaled beyond the point where a single server can provide adequate processing power. Both enterprises and service providers need the flexibility to deploy additional servers quickly and transparently to end-users. Network load balancing makes multiple servers appear as a single server – a single virtual service – by transparently distributing user requests among the servers.

There are many websites which are currently using this system. Network Load Balancing is implemented through the use of clusters on Microsoft’s Server 2000. Many load distributing algorithms have been used to distribute the load among servers. Weighting, Randomization, Least connections, Minimum misses, Hashing etc have been used by many load balancing systems. Furthermore, environments assumed by many of the algorithms have different features on task scheduling than that of a workstation-based locally distributed system. For example, many studies of load balancing algorithms assume that all the sites in the system are subject to the same average arrival rate of tasks, i.e., over the long term, the external load imposed on each site is the same. Today many load balancing systems are using these algorithms for distributing load among clusters and within clusters among many servers.

**Proposed System**

The system which we are proposing is Network Load Balancing. In this system when a client sends a request to the server then a machine called load balancer will firstly check that whether the load has increased or not. If not, then the client’s request will get served. And if load get increased load balancer would use an algorithm to distribute the load among servers. In this system we are using clusters. Clusters can be made either by using Microsoft’s Server 2003 or above. The algorithm which we are using for load distribution is Round-Robin DNS (RR-DNS). This method has proven quite effective in real-world scenarios. The round robin algorithm can be effective for distributing the workload among servers with equal processing capacity. When servers differ in their processing capacity, using response times or number of active connections as the selection criteria can optimize user response times.

The System can be used by all the websites where load balancing problem is the major concern. It is used for making the user able to access the site at any time. This system can also be used by an organization where the internal network is suffering from the problem of load balancing. Network Load Balancing does not need any specialized user. The user should have the knowledge of accessing the internet applications only. This system also hides the internal operation of balancing load among servers from the user. So user will not have to worry about the load problem.

**Limitations**

The limitations of the various Network Load Balancing Systems are:-

- The algorithms that are used in load balancing systems such as Weighting, Minimum misses, Hashing, Least connections etc are not effective.
- These algorithms are used where no of servers to distribute the load are less.
- Microsoft server 2000 is now become the old version for making the clusters.

**Functions**

The client sends its request to the server. The request is received by the load balancer which is a computer system. The load balancer then checks the server.
Design

- **Use case Diagram**
  Use case diagram represent the overall scenario of the system. A scenario is nothing but a sequence of steps describing an interaction between the user and the system.

**Class Diagram**
Class diagram describes the static structure of the various classes of the system and their association but the classes declared with the system, which does not declare how the class, behaves when they act economically.
Conclusion

Many content-intensive applications have scaled beyond the point where a single server can provide adequate processing power. Both enterprises and service providers need the flexibility to deploy additional servers quickly and transparently to end-users. Network load balancing makes multiple servers appear as a single server – a single virtual service – by transparently distributing user requests among the servers. The system which we are proposing is Network Load Balancing. In this system when a client sends a request to the server then a machine called load balancer will firstly check that whether the load has increased or not. If not, then the client’s request will get served. And if load get increased load balancer would use an algorithm to distribute the load among servers. In this system we are using clusters. Clusters can be made either by using Microsoft’s Server 2003 or above. The algorithm which we are using for load distribution is Round Robin DNS (RR-DNS). This method has proven quite effective in real-world scenarios. The round robin algorithm can be effective for distributing the workload among servers with equal processing capacity. When servers differ in their processing capacity, using response times or number of active connections as the selection criteria can optimize user response times.

References