Redundancy Vs Replication

Redundancy : In this the nodes/servers are having their different copy. This can be list of Server having the case code base/jar file that responds to the incoming requests. Types : Active and passive.

Active : here all the severs are active and respond to the incoming request.
Passive : here the sever that are copy are stand by and if the main/active server goes off then only they became active to process the request.

Replication : In this we have redundancy as well as synchronisation, this is basically used in databases as we have database server we not only have the server , we also have servers withe data in sync with all of them.

Active : here all the severs are active and have the in-synced data on all of them.
Passive : here the severs, be haves as Master and slave where master gets the updated data and it updates the data to its slave via Synchronous/Asynchronous process using scheduler.

LoadBalancer

LoadBalancer is an efficient way of distribution of the incoming network traffic across all the distributed systems.

Roles of a LoadBalancer :

- The load distribution is equal over every node.
- Health Check(if the node is not running, the requests are passed to another nodes).
- It ensures high scalability, high throughput and high availability.

Challenges of a LoadBalancer :

- Single point of failure, to overcome this we need to add on a passive load balancer.

Advantages :

- Optimisation
- Better user experience
- Prevents Downtime
- Flexibility
- Scalability
- Redundancy

Load Balancing Algorithms:

- **Round Robin (Static)** : Rotation old fashion.
- **Weighted Round Robin (Static)** : It is similar to Round robin, if the server are of different capacities.(some node may of better resource and same may not).
- **IP Hash Algorithms (Static)**: All server have almost same capacity, and the hash function (input is source IP) is used to random or unbiased distribution of request to the nodes.
- **Source IP Hash (Static)**: It combines the client and source IP addresses to produce a hash key. The key can be used to determine the request distribution.
- **Least Connection Algorithms (Dynamic)**: Client request is distributed to application servers with least number of active connections at the time the client request is received.
- **Least Response Time (Dynamic)**: The request is distributed to servers which has the least response time.