# Load Balancing

Load Balancing is a key concept to system design. One simple way would be hashing all requests and then sending them to the assigned server.

The standard way to hash objects is to map them to a search space, and then transfer the load to the mapped computer. A system using this policy is likely to suffer when new nodes are added or removed from it.

Some terms you would here in system design interviews are Fault Tolerance, in which case a machine crashes. And Scalability, in which case machines need to be added to process more requests. Another term used often is request allocation. This means assigning a request to a server. Load balancing is often tied with service discovery and global locks. The type of load we want to balance is that of sticky sessions.

Consistent Hashing allows requests to be mapped into hash buckets while allowing the system to add and remove nodes flexibly so as to maintain a good load factor on each machine.

The standard way to hash objects is to map them to a search space, and then transfer the load to the mapped computer. A system using this policy is likely to suffer when new nodes are added or removed from it.

Consistent Hashing maps servers to the key space and assigns requests(mapped to relevant buckets, called load) to the next clockwise server. Servers can then store relevant request data in them while allowing the system flexibility and scalability. If a node is removed only the data mapped in that node is affected while the other nodes remain unaffected. The data in the node that is removed is moved to the next immediate bucket node.