Hash function takes a key of arbitrary length as input and return a hash value of fixed length. If the nodes/servers/hash tables are of fixed length then we can use mod hashing. Mod hashing will mod the hash value with the number of nodes.

We can not use mod hashing if the number of nodes are not fixed because when the number of nodes changes for the same key it will give different value. So we will not find the data. To get the data we have to do rebalancing the data from one node to another node. If we use mod hashing we have to do rebalancing for all the db entries which is inefficient.

In load balancer and horizontal sharding, number of nodes are dynamic. In load balancer, application server can crash or we can add new application server. In horizontal sharding, db servers can fail or we can add new db server. So we can not use mod hashing here. We will use consistent hashing for these.

We have to do consistent hashing in such a manner that the number of rebalancing should be => (1/n)% of total number of keys. Here, n= number of nodes and %= percentage. Consistent hashing uses virtual ring. It also use mod hashing to determine the position of the servers in the virtual ring. In this mod hashing we will use the total value of positions in the virtual ring instead of the number of nodes. Lets assume servers are at the same distance on the virtual ring. So there is high chance of equally distributed data. We can use mod hashing to determine the position of the key on the virtual ring. To determine which key belongs to which server we will use clockwise rules. If we move the key clockwise then the first server will be the belonging server.

if the servers are not in same distances on the virtual ring, then there is high chance of data not being equally distributed. To solve this problem we can replicate the servers at the different positions of the virtual ring. Then we will follow the clockwise rules. So there will be high chances of data being equally distributed. We have to replicate the nodes as many times as the rebalancing need to rebalance (1/n)% of total keys.

====================  
Questions  
====================  
1. How to get the position of the server on the virtual ring?