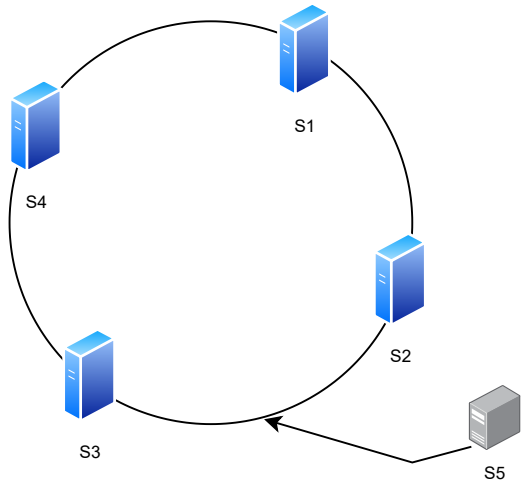
Problem with normal Hashing:

To add or remove a node we need to rebalance/move lots of data to keep the hash work

Consistent Hashing:

- For each server get a hash value using a hash function for a ring and put the server in the ring based on that hash value.
- Replicate the data to next server in the ring based on replication factor. So every server will contain it's own data and replica of other server.
- Use the same hash function to find a hash value on the ring and put the data in next server in the ring



> To add a new server (eg: between S3 and S2 based on hash) we only need to copy data for range (S2->S3) from S3 to S5.

> For a node failure all request to that node will be handled by next node (it already has replica of failed node)

Uneven Distribution Problem:

Solution: Virtual Node (using different hash function).

eg: Each server has 3 virtual nodes.