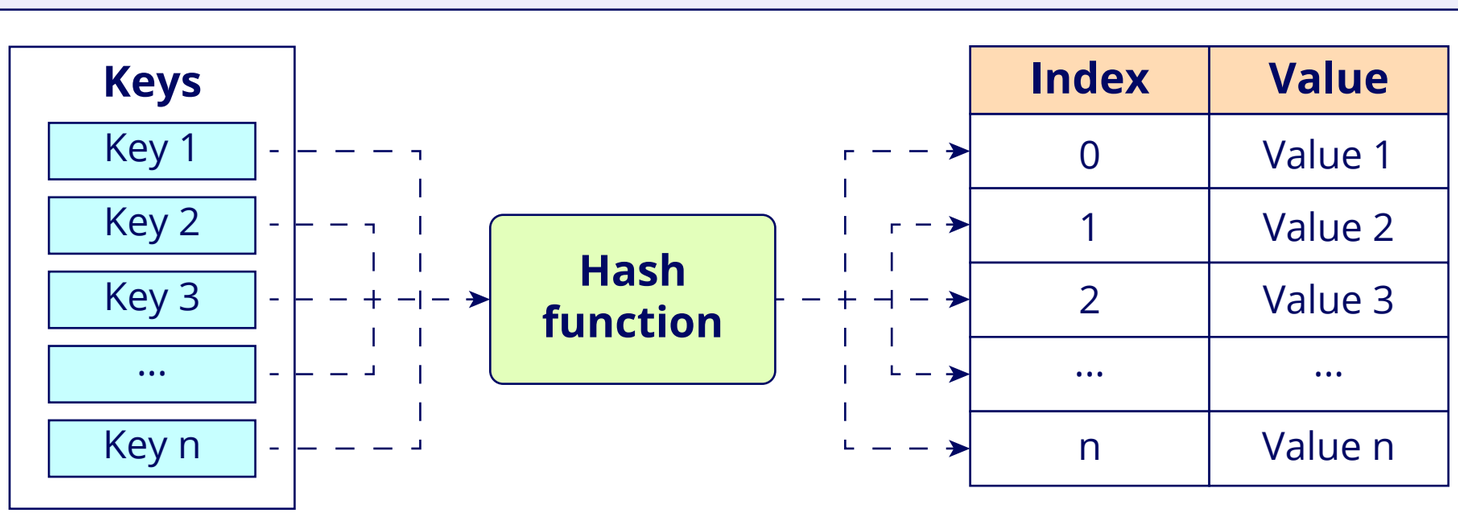
What is a Hash Function?

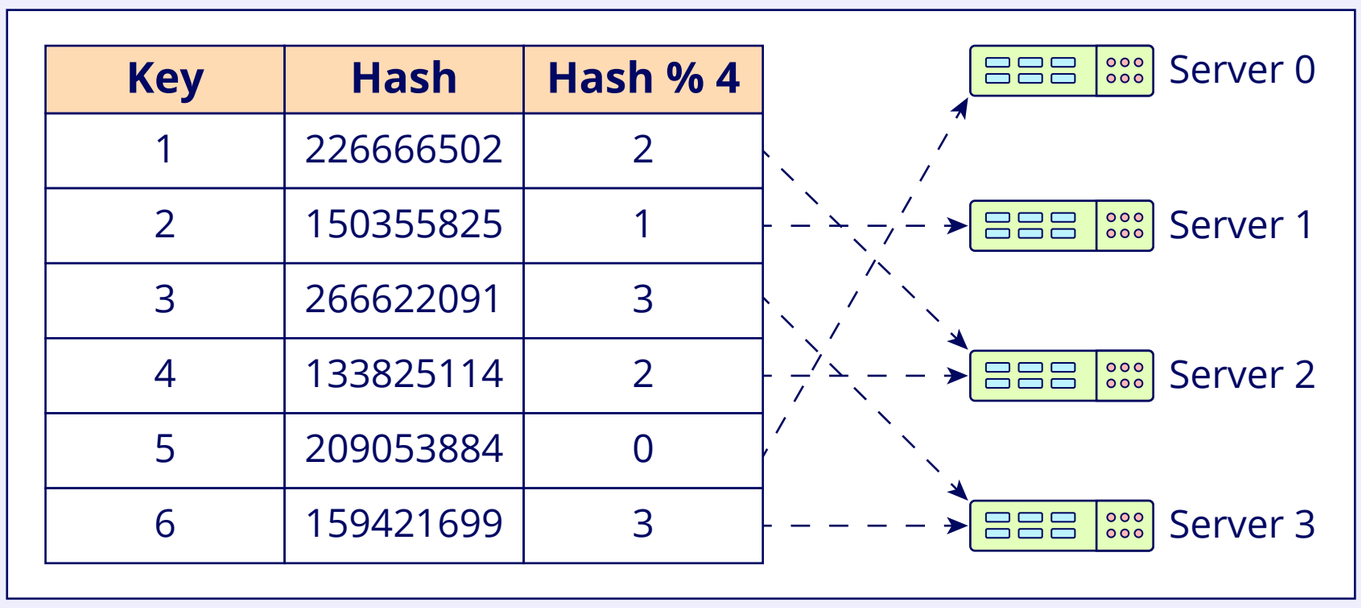
A hash function takes an input (key) and produces a fixed-size string of bytes (hash code or hash value). This hash code is used to index data in hash tables, enabling efficient data retrieval and lookup operations.



***Note:** A good hash function aims to distribute keys uniformly across the hash table to minimize collisions.

Distributed Hash Tables

Often, the size of a hash table becomes very large; this means it has to be split into several parts. Each part of the hash table is stored on a different server, as shown below:



Key	Hash	Hash % 4
1	226666502	2
2	150355825	1
3	266622091	3
4	133825114	2
5	209053884	0
6	159421699	3

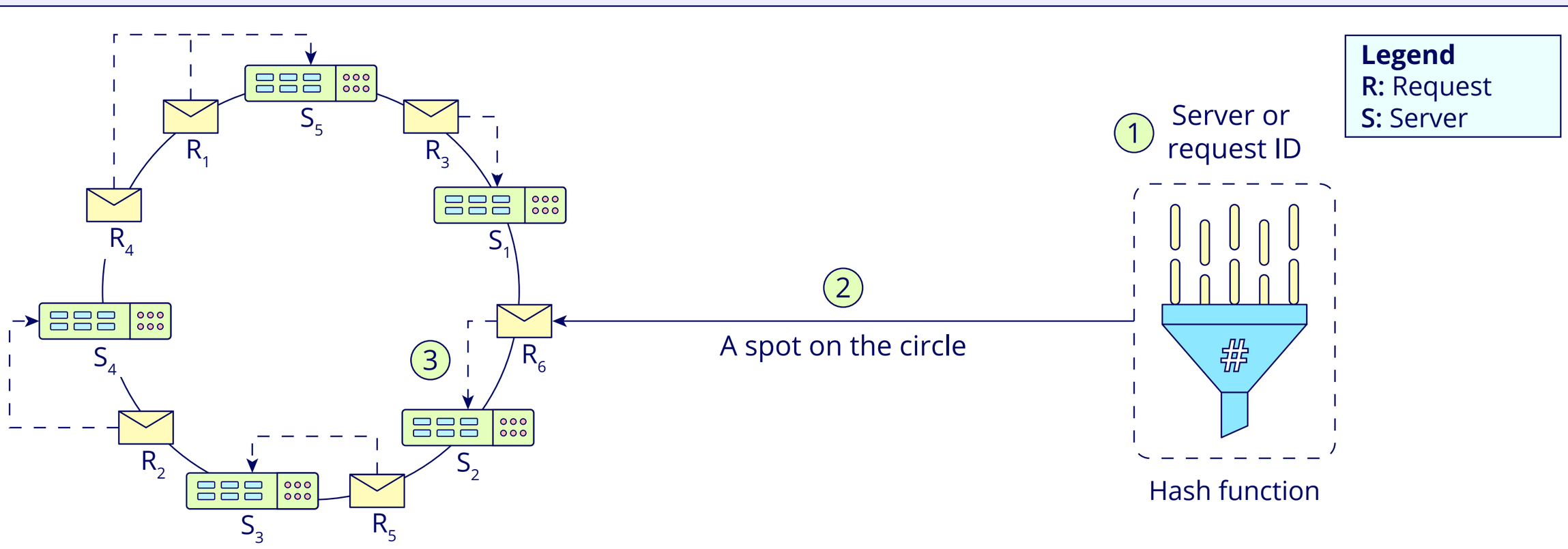
Problem!

If the number of servers is reduced in the cluster, the modulo must be recalculated accordingly for each value (an expensive operation!).

Solution: Consistent hashing

What is Consistent Hashing?

Consistent hashing is an effective technique for distributing the workload among a set of servers efficiently placed on an imaginary circle. It minimizes the number of keys that need to be remapped when a server is added or removed from the cluster.

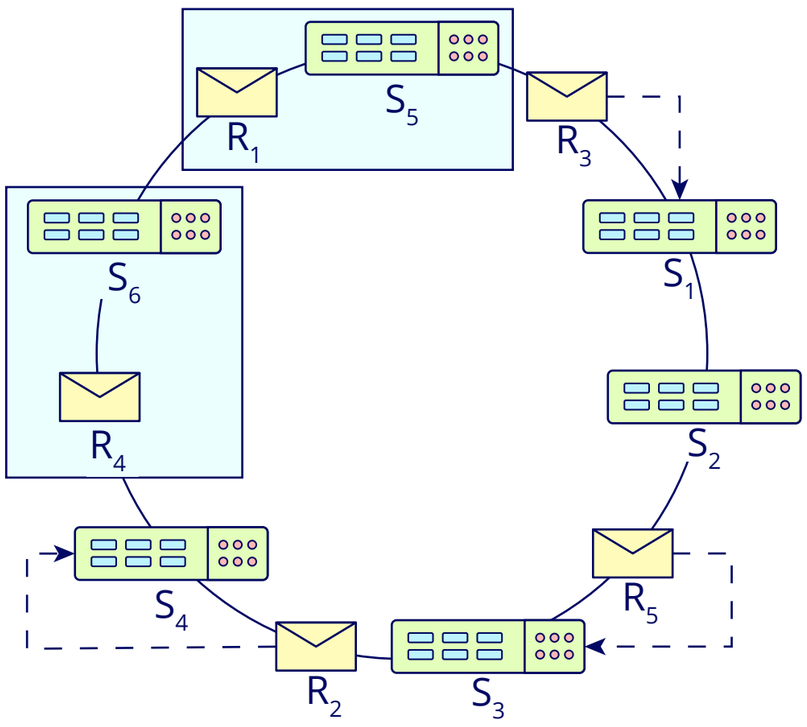


Working of consistent hashing: Each request (R) is assigned to the nearby server (S) on the circle.

Note: Different types of data are stored on various servers (1–5), and the same hashing operation is used for retrieval. This demonstrates how hashing performs load balancing.

What is Consistent Hashing?

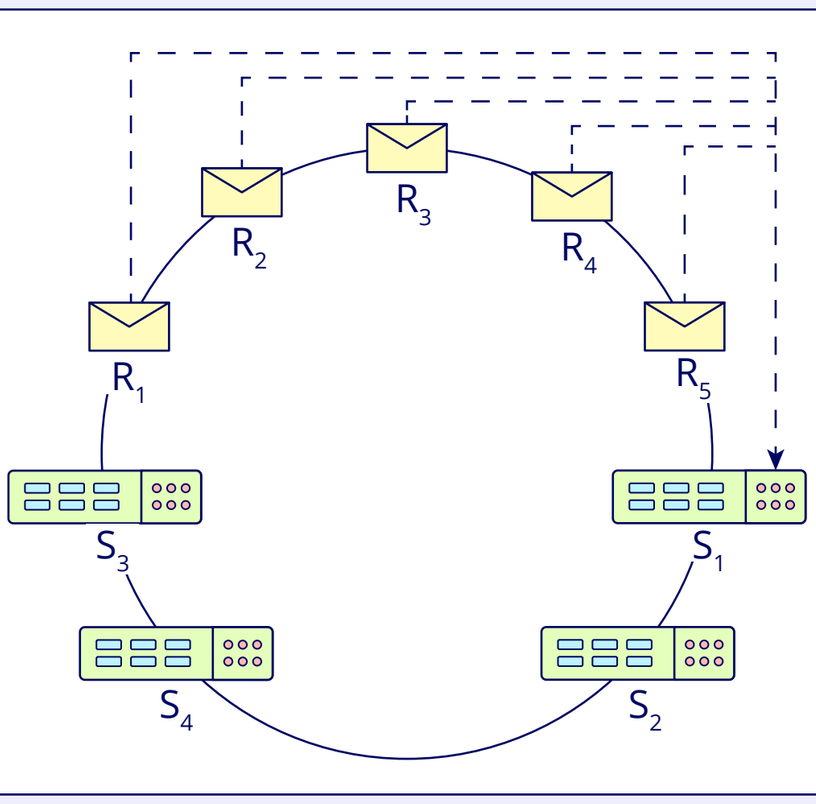
Adding new servers reduces the burden on other servers in the cluster using consistent hashing.



Impact of adding a server to the existing circle

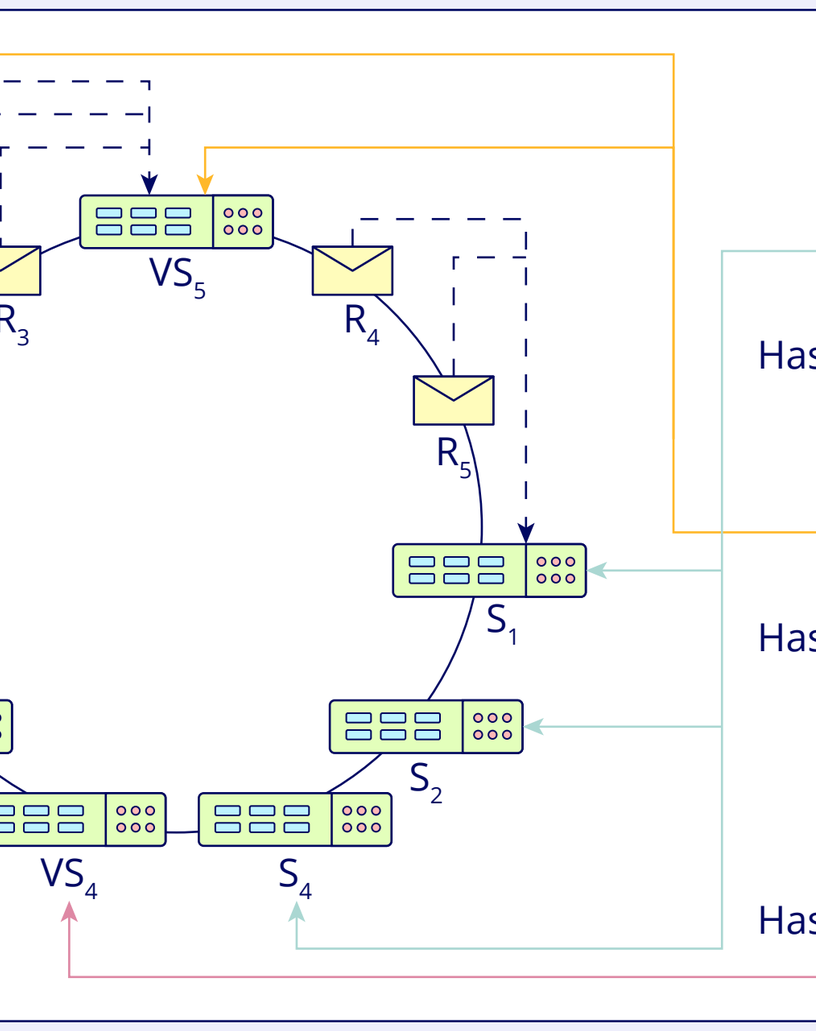
What is a Hotspot in Consistent Hashing?

As depicted below, most of the incoming requests lie between the S_3 and S_1 servers. Consequently, S_1 has to handle most of the requests compared to other servers, becoming a hotspot.



Hotspot Prevention: Use Multiple Hash Functions

To prevent hotspots, we use virtual servers (see VS_2 and VS_4) for balanced workload distribution. Virtual servers map each physical server to multiple locations using multiple hash functions.



Some important use cases of consistent hashing

- Web caching and CDNs
- Distributed databases
- Sharded counters
- Distributed file systems