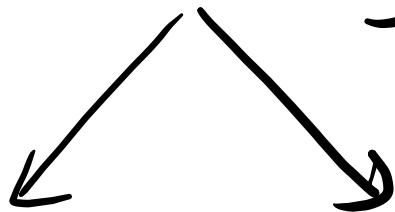


Consistent Hashing

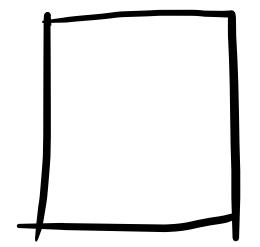
Amazon DynamoDB

Apache Cassandra

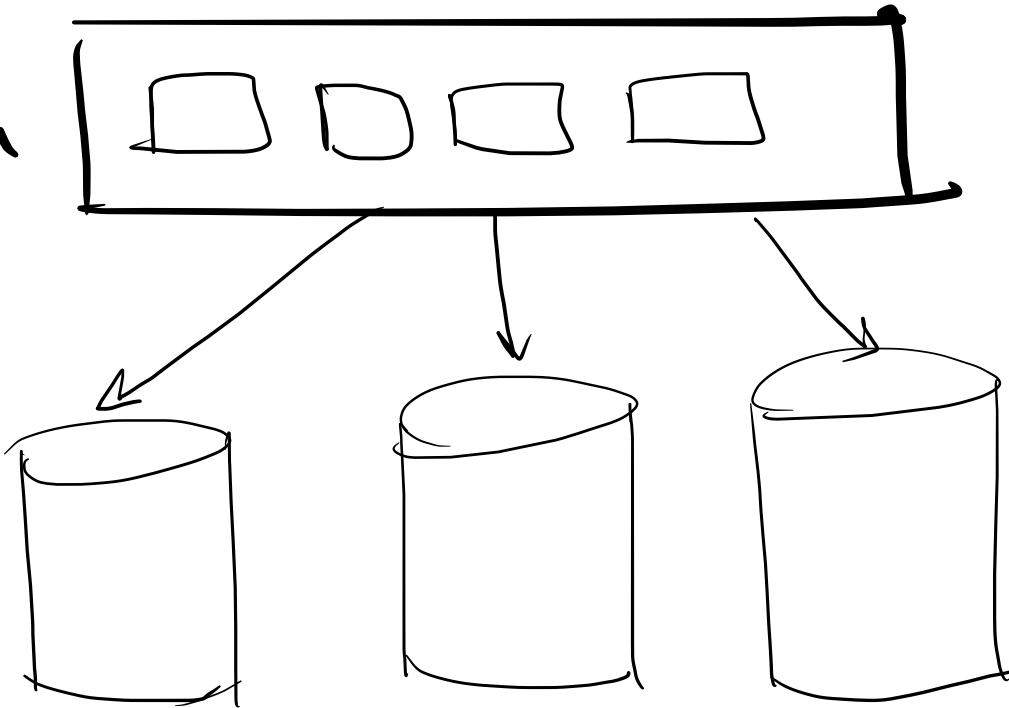
Scaling



horizontal vertical

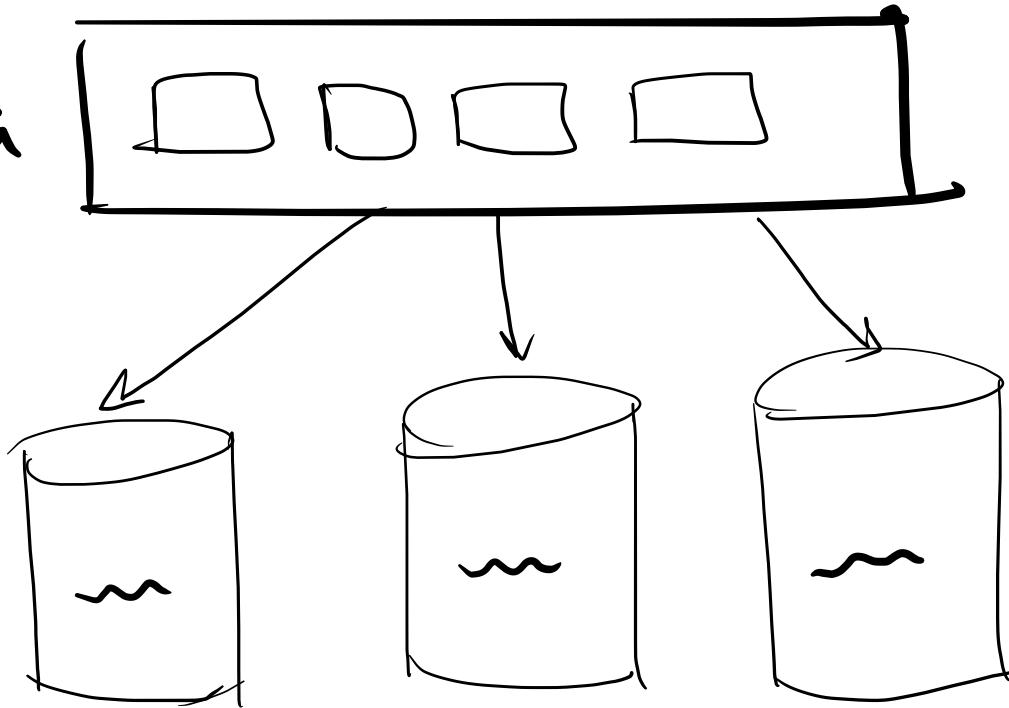


Data



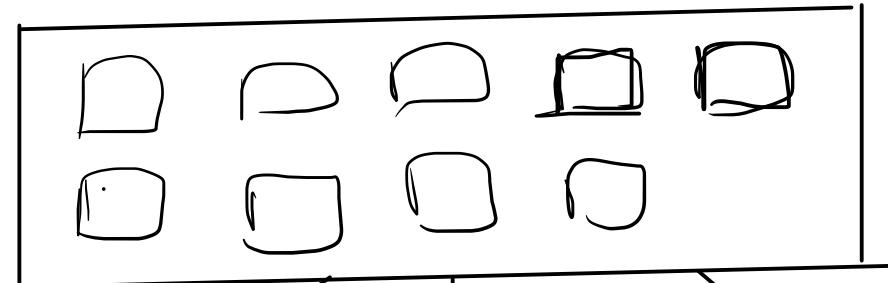
Horizontal Scaling

Data



Horizontal Scaling

Data



Data
distributed
evenly
across
servers

Hashing

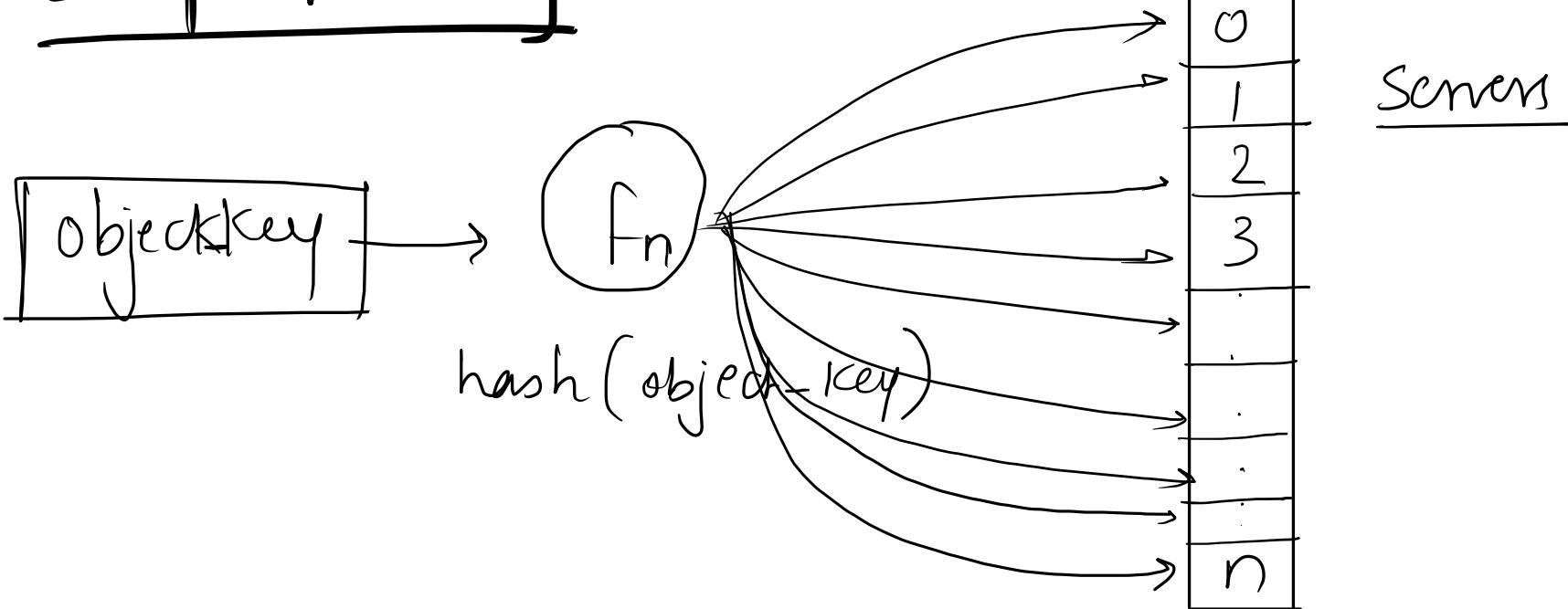
①

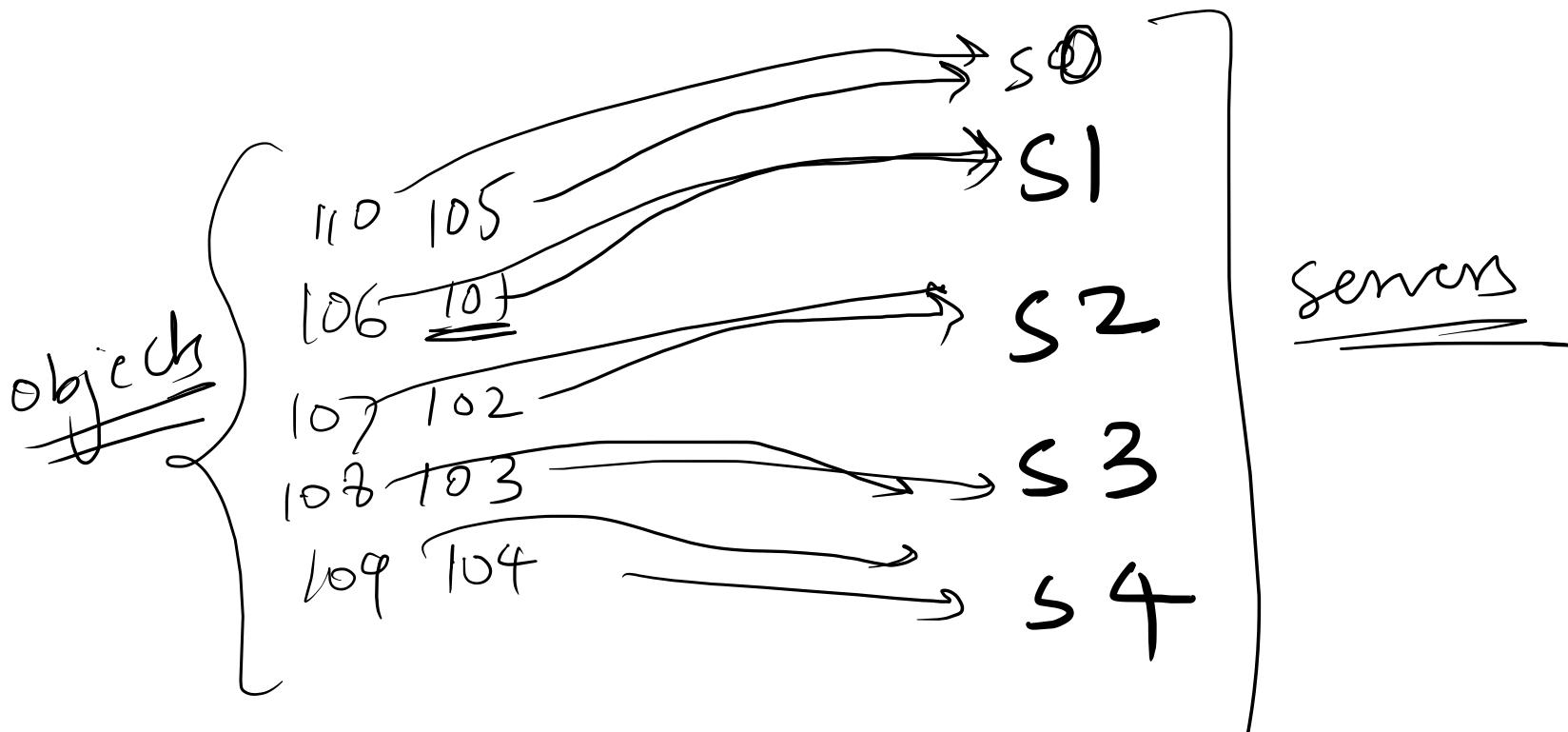
Simple Hashing

$\text{serverIndex} = \text{hashkey \% N}$

$N \rightarrow$ Size of server pool

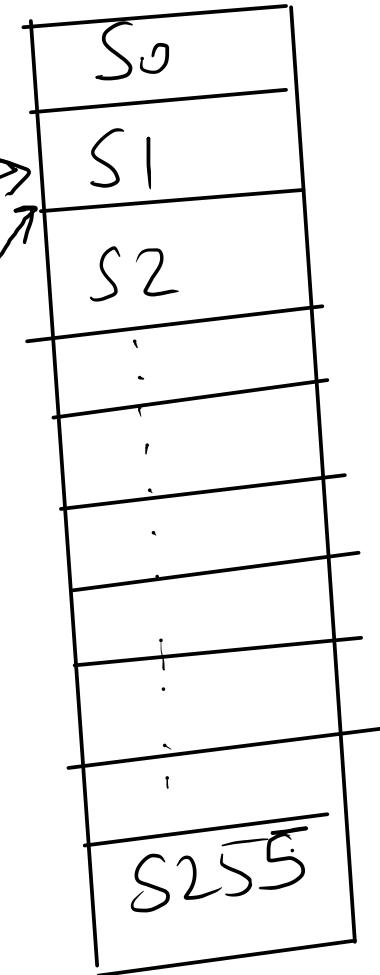
Simple Hashing → even distribution





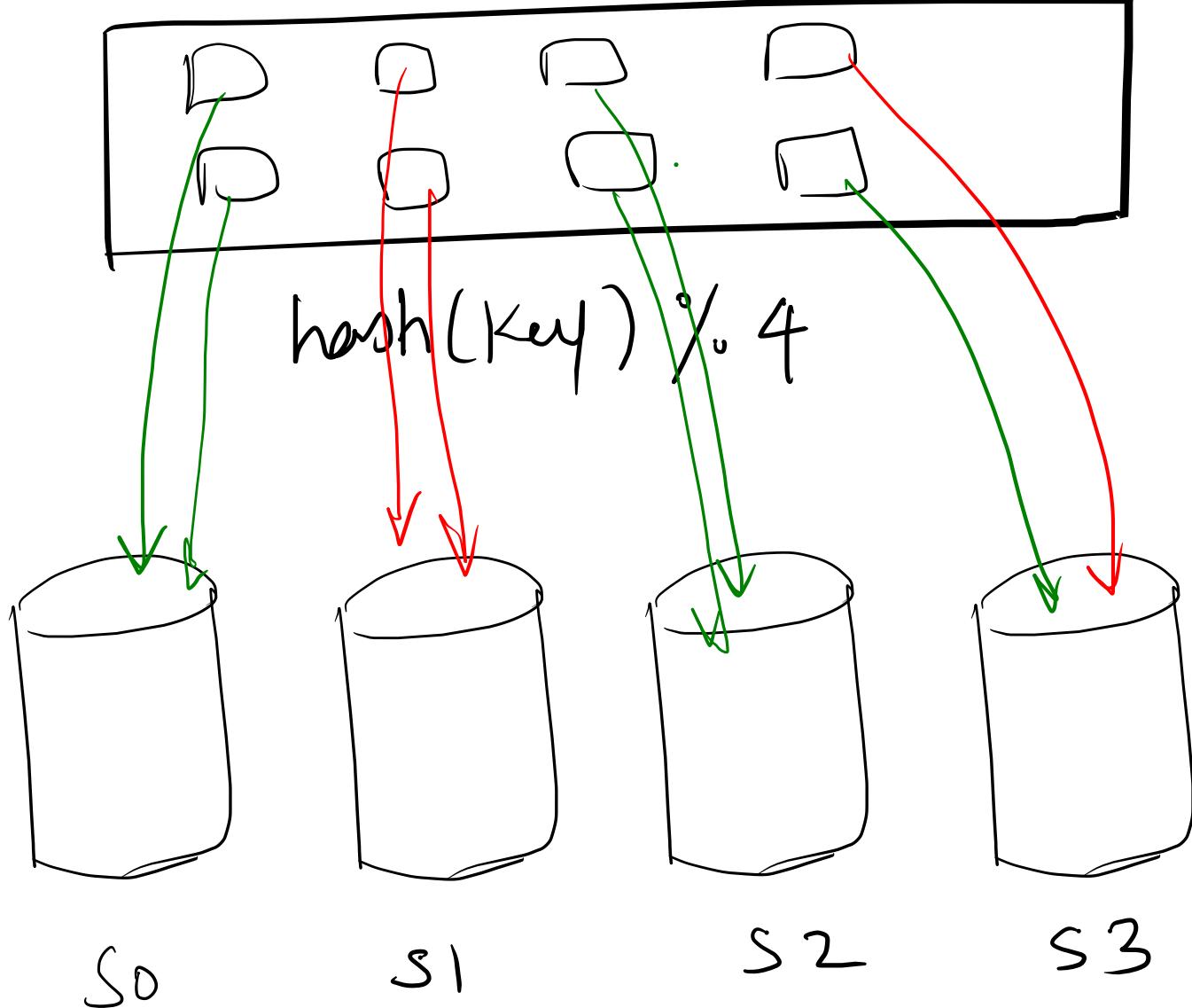
~~hash(object-Key)~~) % 2⁵⁶

hash(object-Key) % 2⁵⁶



If number of servers remain
same, object key will be mapped
to the same server.

Data

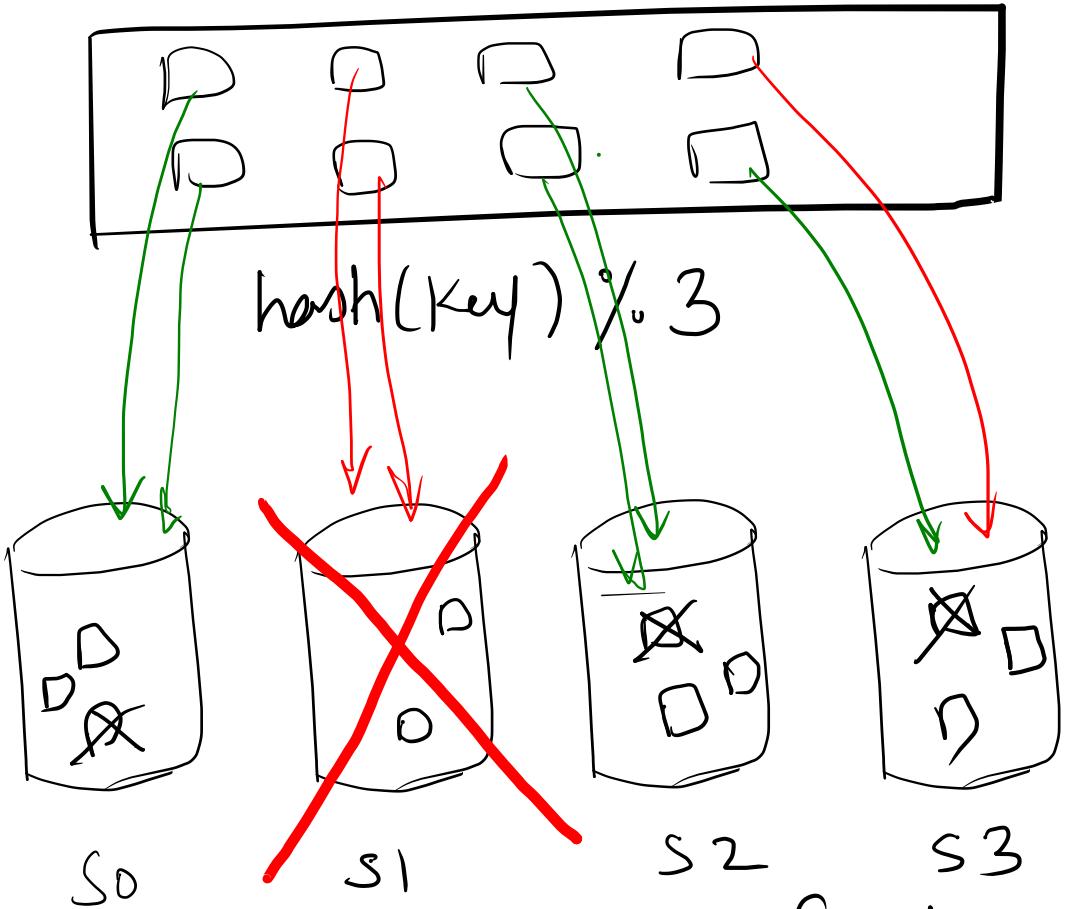


Simple Hashing

If server is removed

Data

- * needs to redistribute the objects
- * lots of objects need to be moved

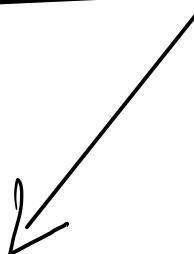


movement of data

Consistent Hashing

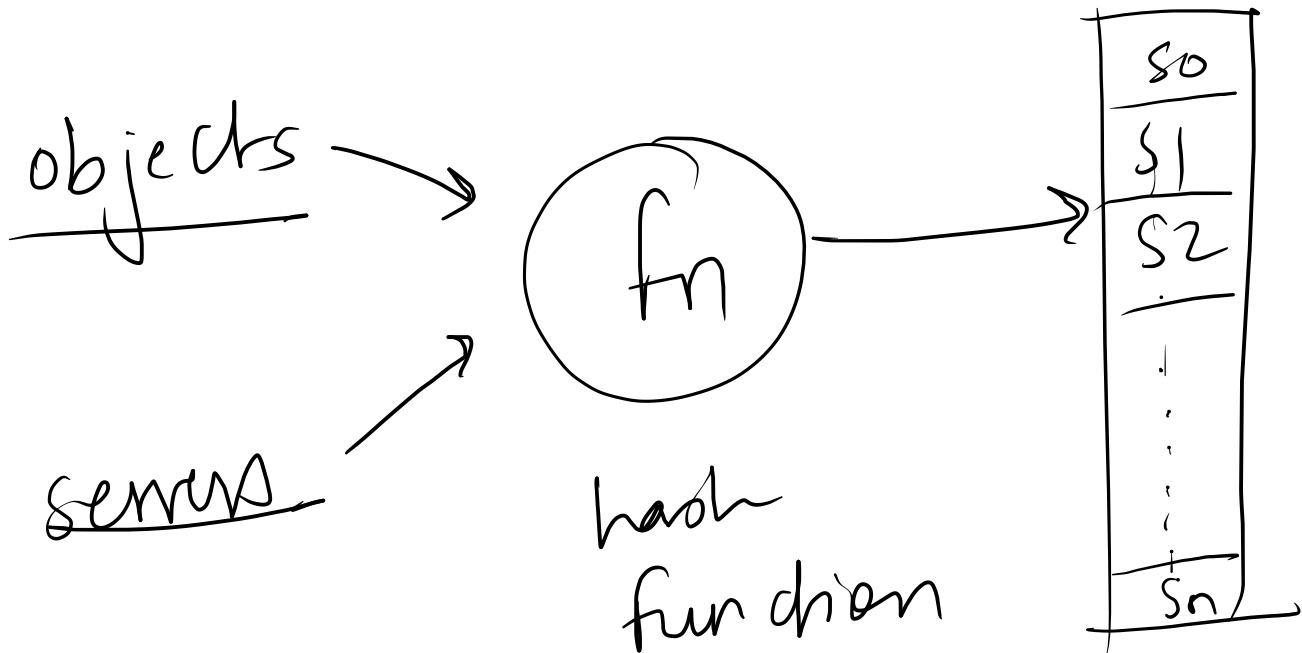
- we want most objects to be assigned to the same server.
- reduce movement of objects

Consistent Hashing



Object Key

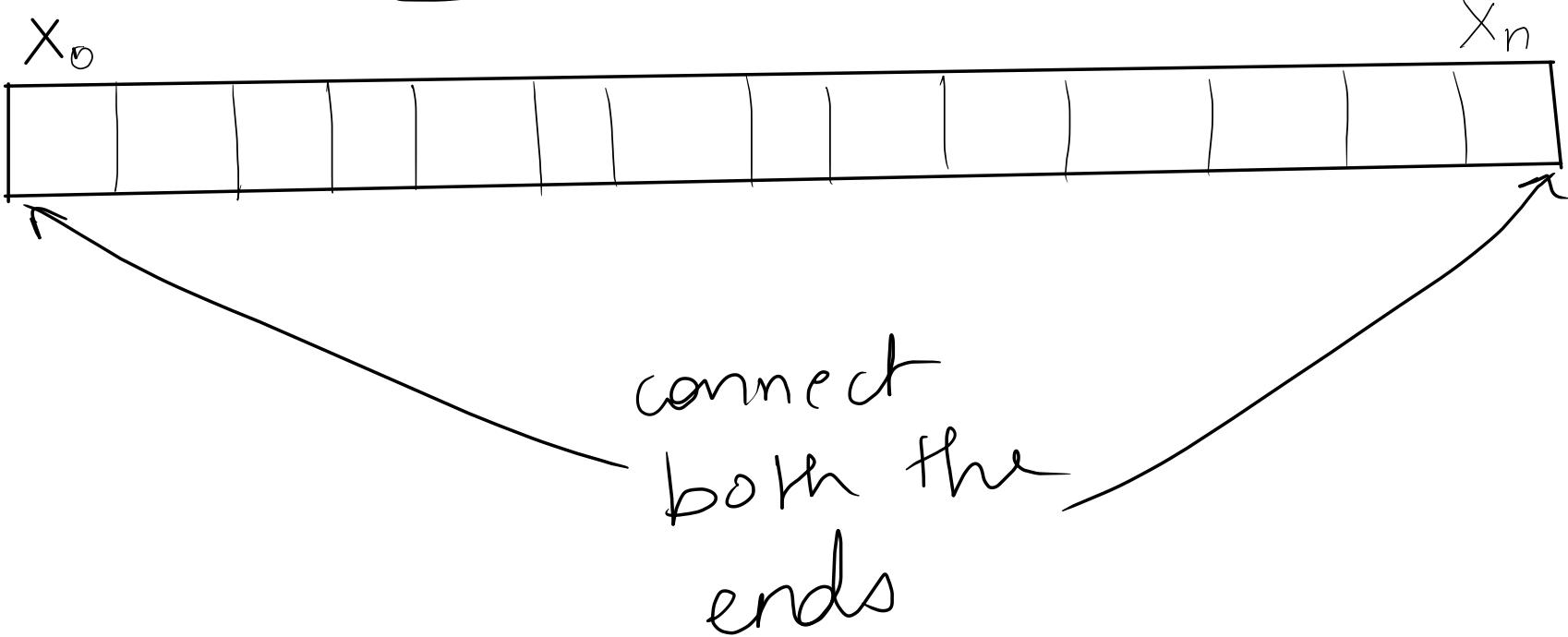
Server Names

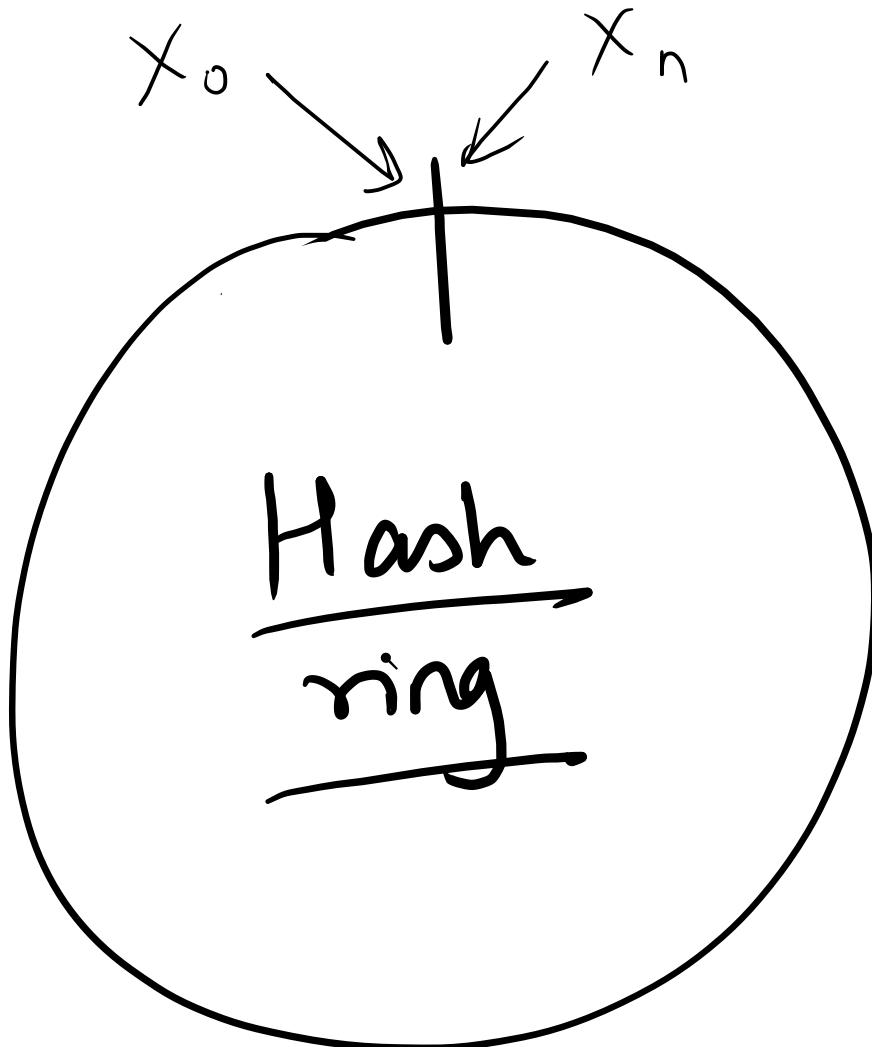


Hash space

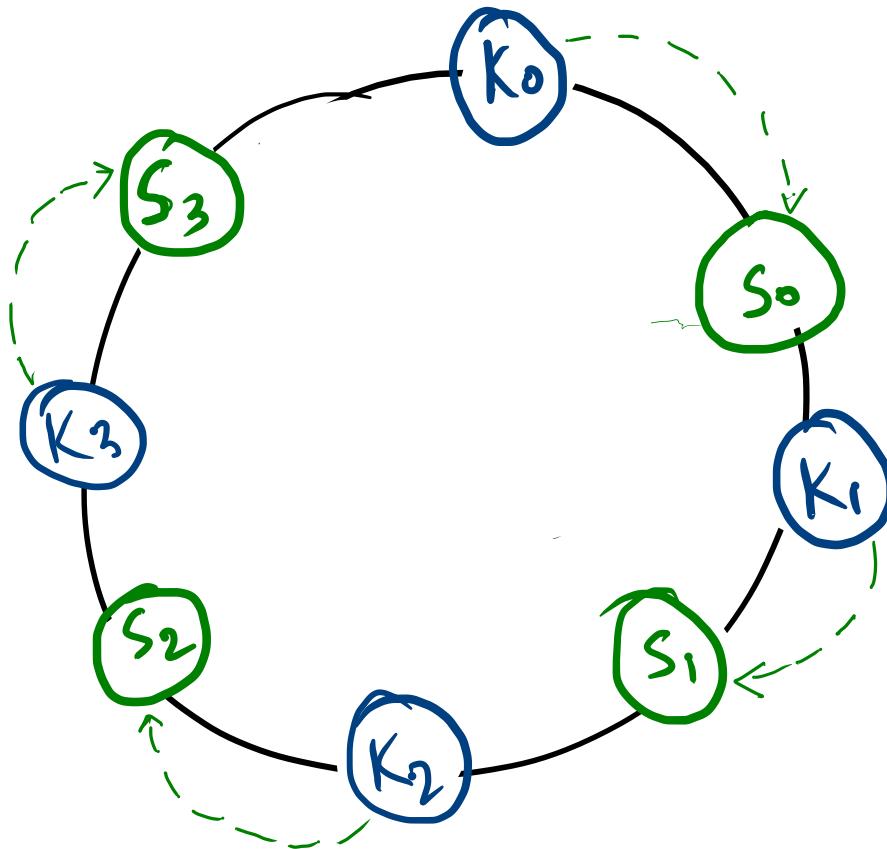
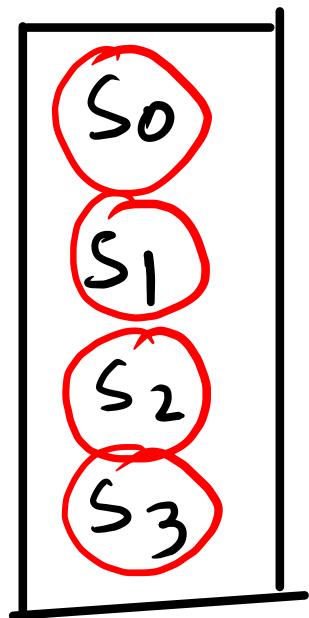


Hash space



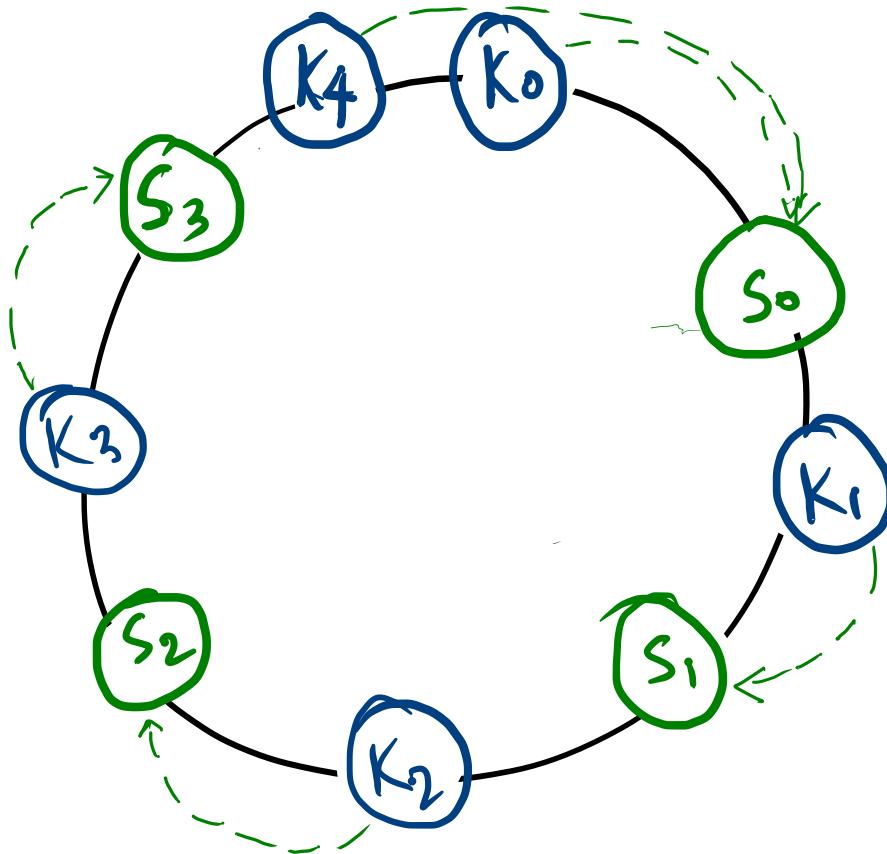
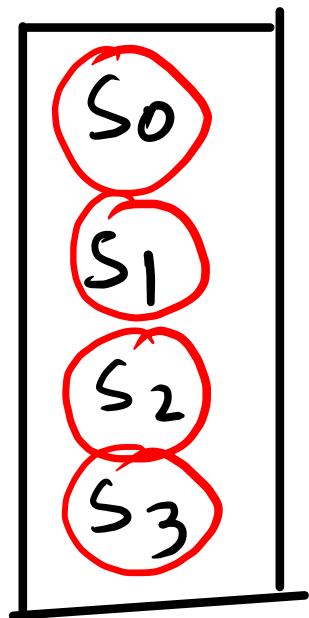


Servers



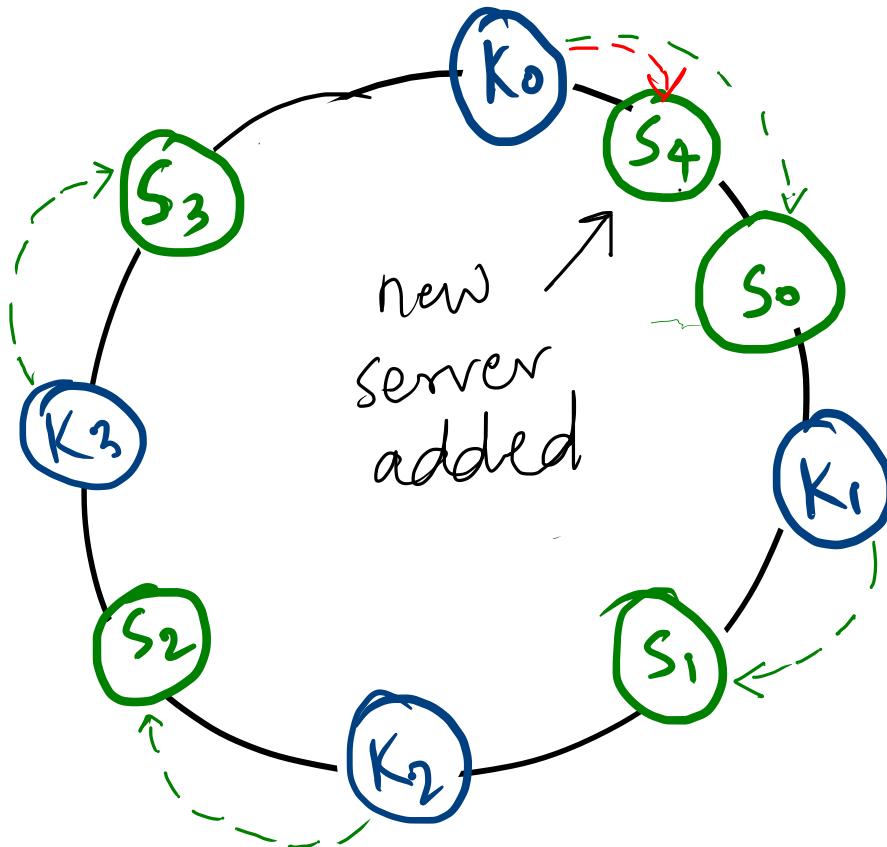
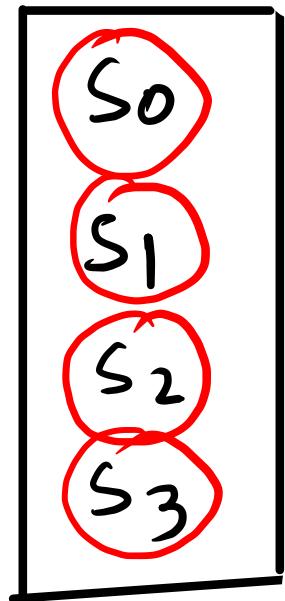
$S \rightarrow$ servers
 $K \rightarrow$ objects

Servers

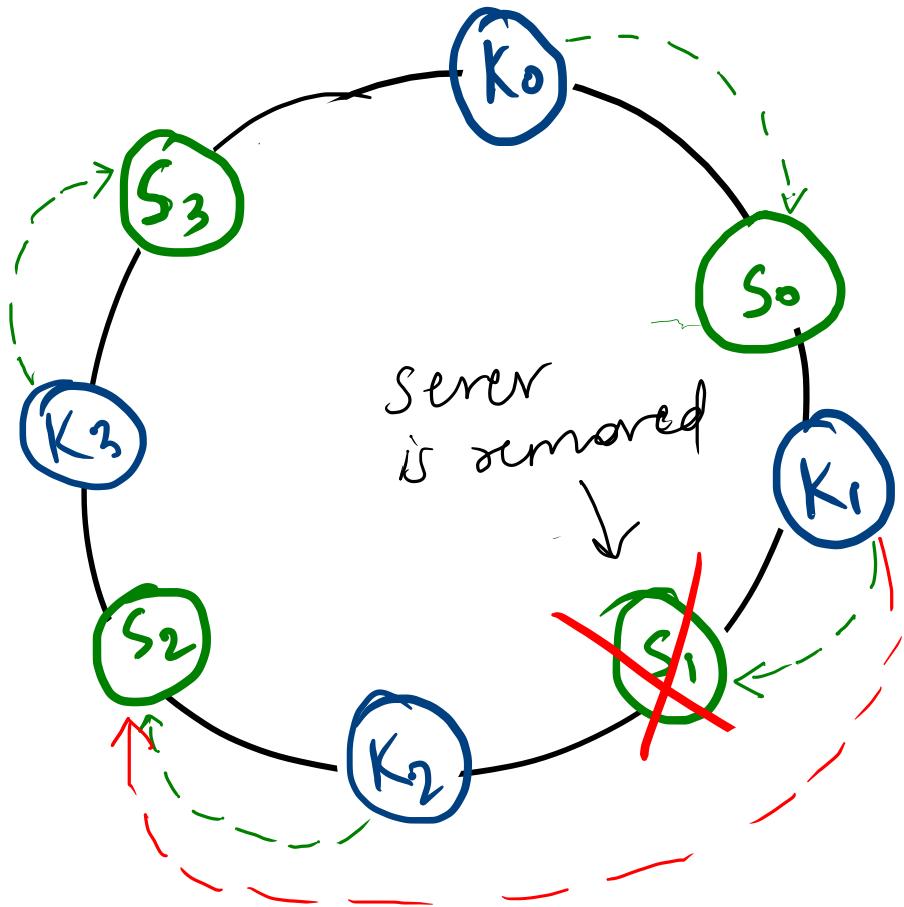
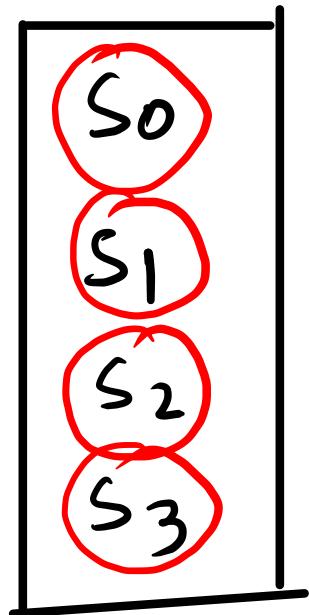


$S \rightarrow$ servers
 $K \rightarrow$ objects

Servers



Servers

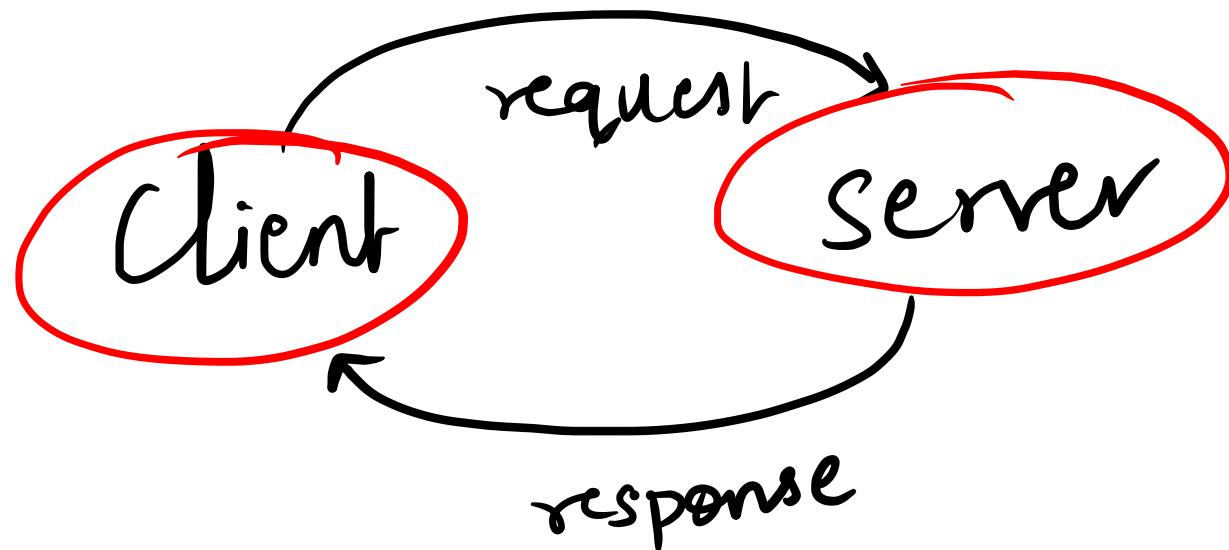


$S \rightarrow \text{servers}$
 $K \rightarrow \text{objects}$

→ to locate the server for an object we go clockwise on the ring from the objects position.

- With simple hashing, when a new server is added, almost all keys need to be remapped.
- With consistent hashing, adding new server only requires the redistribution of fraction of keys.

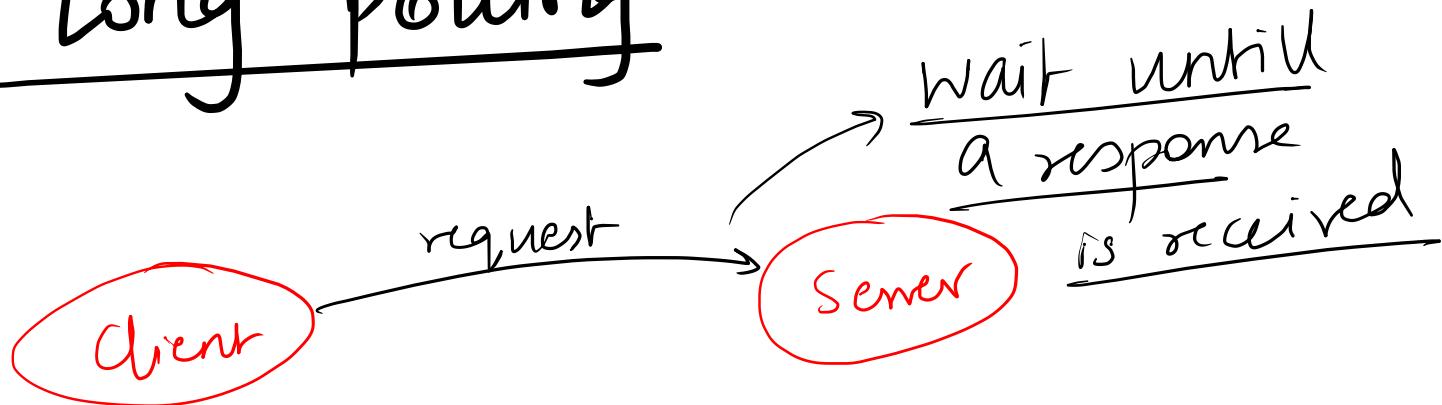
Communication Protocols



- ① long polling
- ② websockets
- ③ Server side Events (SSE)

- Real-time data transfer b/w client & server
- receive updates from server in real-time
- more interactive & responsive web applications

① Long polling :

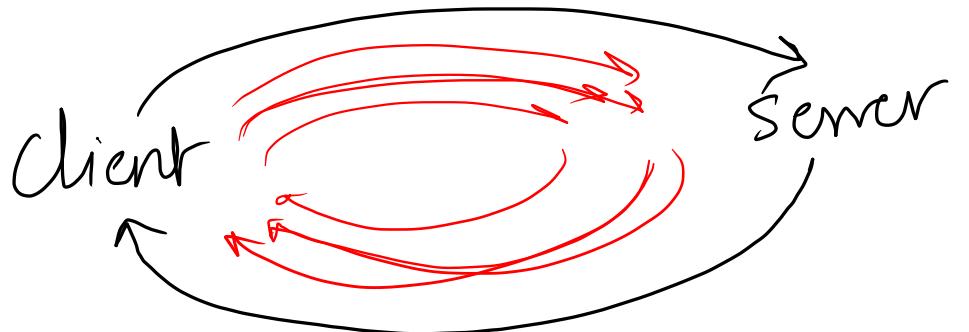


- 1.) sends a request to server
and wait until response is received
→ one-way communication
- 2.) the connection is active, till it
has information to send back
to the client

Long Polling

- 1.) client sends a request , waits for response
 - 2.) server keeps the connection open ,
until it has data to send back .
 - 3.) once the client receives the response,
it sends another request to the server
- process repeats

② Websockets



* Connection stays active until closed by client/server.

Communication in both directions

Websockets

- 1.) Client and server both can communicate, in both directions.
- 2.) Using websocket protocol
- 3.) Data can be delivered and received simultaneously, without further request.
- 4.) Connection stays active, until closed by client/server.
- 5.) Client no longer needs to make requests to the server regularly.

Websockets

- ① Customer support chat
- ② Data Broadcasting

Server-sent Events (SSE)

- allows a webpage to get updates from a server.
- Server-sent event is when a webpage automatically gets updates from server.

Server-sent Events (SSE)

- allows a webpage to get updates from a server.
- Server-sent event is when a webpage automatically gets updates from server.

Examples

- ① Stock Price Updates
- ② sports result
- ③ facebook / twitter updates