# Consistent Hashing

Jooho Lee, Jay

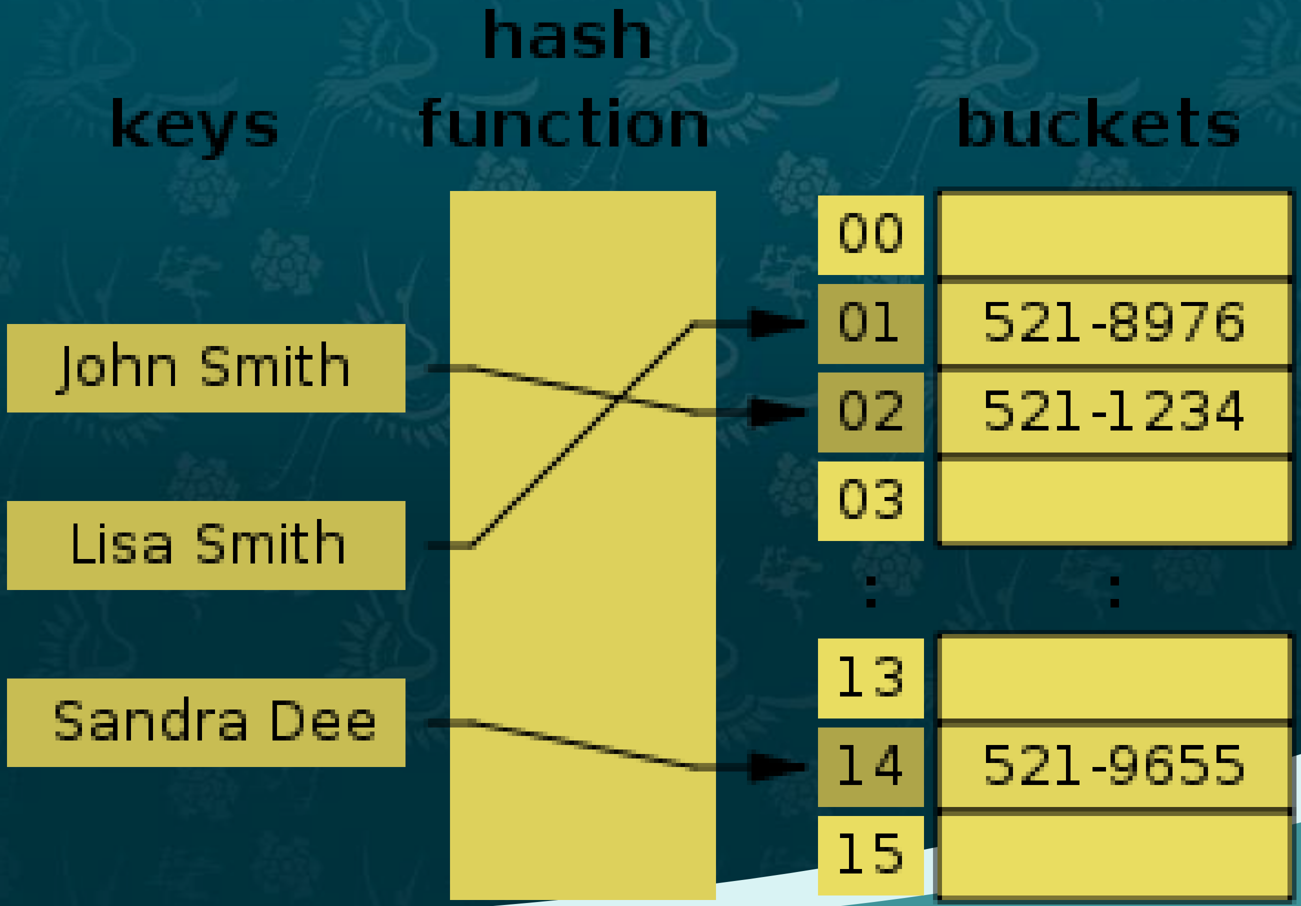
# Agenda

- 1. What is Hash?
- 2. Why we need another Hash Algorithm?
- 3. Fundamentals of Consistent Hashing
- 4. Strength of Consistent Hashing

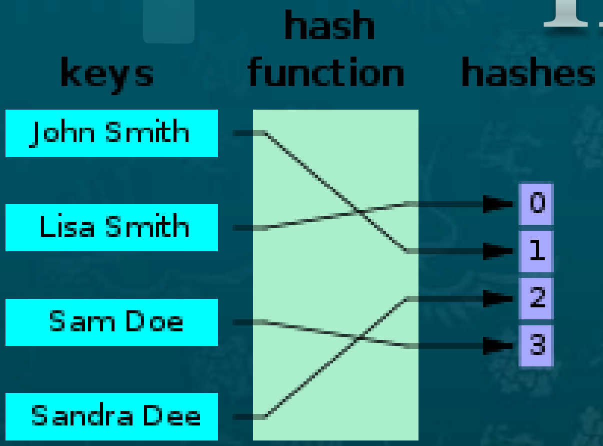
# What is Hash?

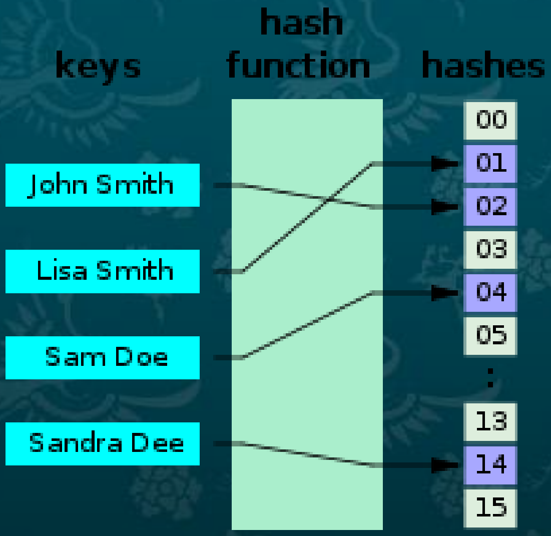
- Hash table
  - Data Structure
- Hash function
  - Calculation

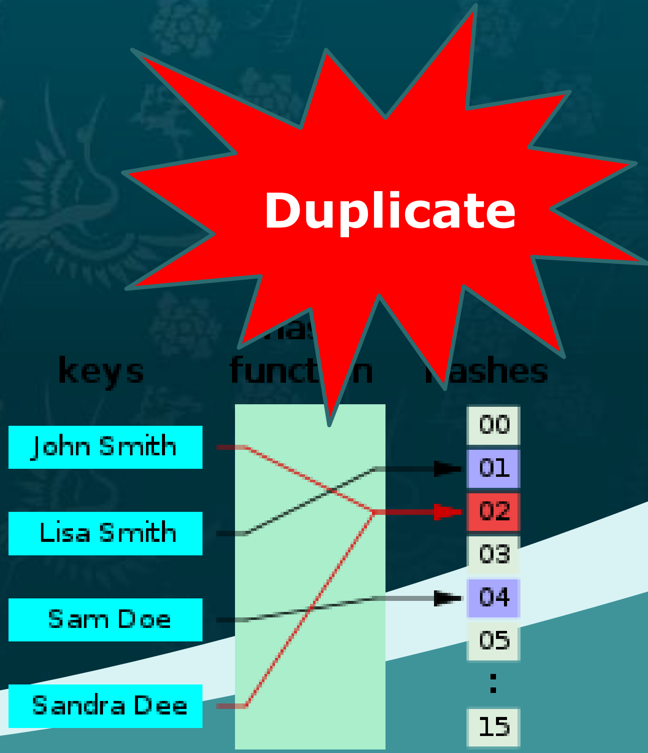
# Hash Table


keys
hash function
buckets
John Smith
Lisa Smith
Sandra Dee
00
01
02
03
:
13
14
15
521-8976
521-1234
:
521-9655

# Hash Function





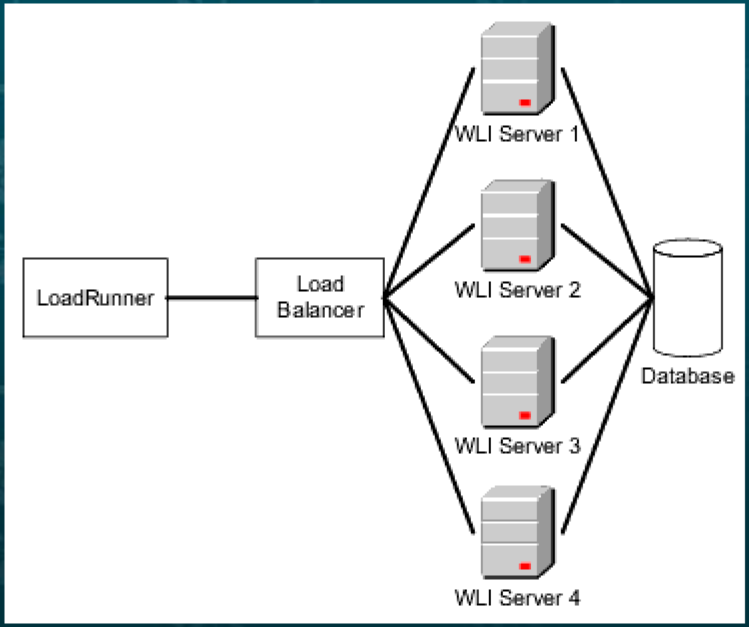


# Why we need another Hash Algorithm?

- Trend
- Poor Point of Original Hashing

# Why we need another Hash Algorithm?

- Trend
  - Cloud Environment
  - Distributed Caching
- Significant Point
  - Horizontal Scalability
  - Capability



# Poor Point of Original Hashing

- Original Hashing Algorithm for Caching
  - (Key k, Value v)
  - Hash Function(k) mod N (machine or size N)

# Poor Point of Original Hashing

- Original Hashing Algorithm for Caching
  - (Key k, Value v)
  - Hash Function(k) mod N (machine or size N)

# Example

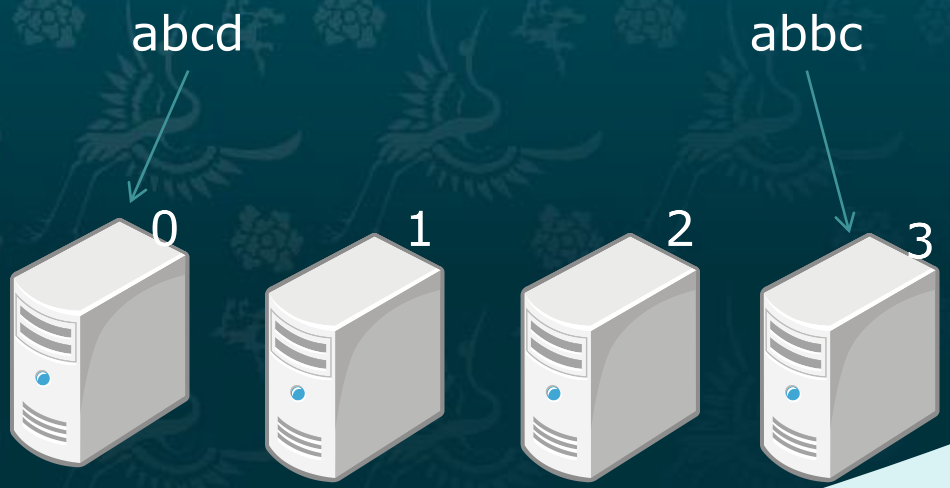
| Binary | 01100001 | | 01100010 | | 01100011 | | 01100100 | |
|---|---|---|---|---|---|---|---|---|
| Hex | 6 | 1 | 6 | 2 | 6 | 3 | 6 | 4 |
| Ascii | | a | | b | | c | | d |

abcd hashes to 0

0x61626364 = 1633831724
16338831724 % 4 = 0

Abbc hashes to 3

0x61626263 = 1633837667
1633837667 % 4 = 3



# If need more machine?

| Binary | 01100001 | 01100010 | 01100011 | 01100100 |
|---|---|---|---|---|
| Hex | 6 1 | 6 2 | 6 3 | 6 4 |
| Ascii | a | b | c | d |

abcd hashes to 0

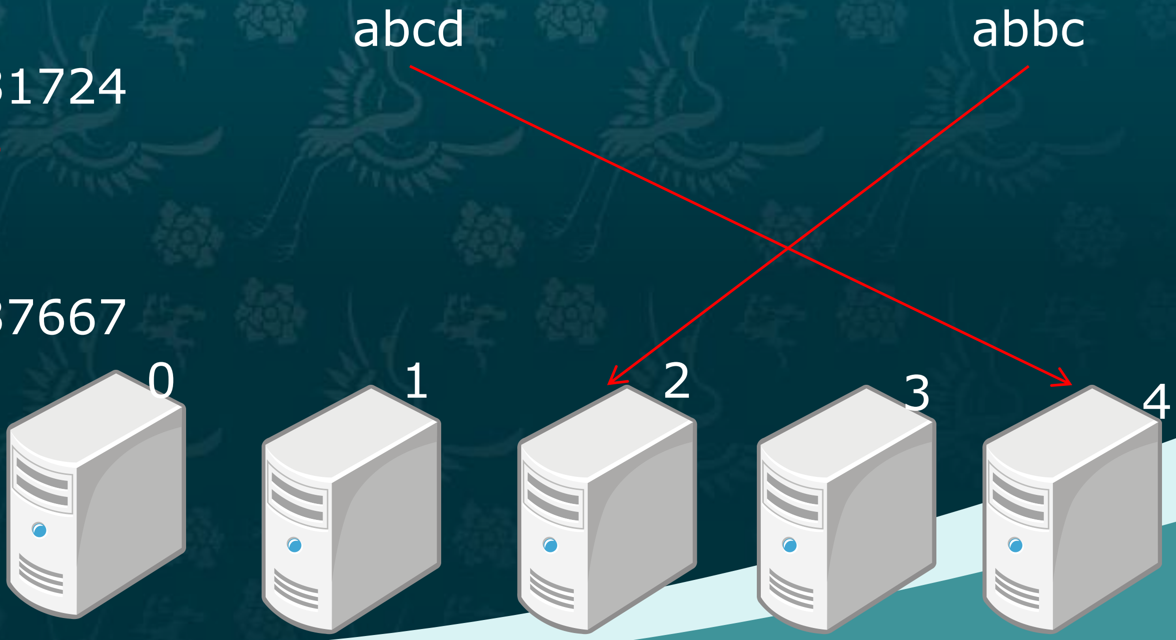
0x61626364 = 1633831724

16338831724 % 5 = 4

Abbc hashes to 3

0x61626263 = 1633837667

1633837667 % 5 = 2



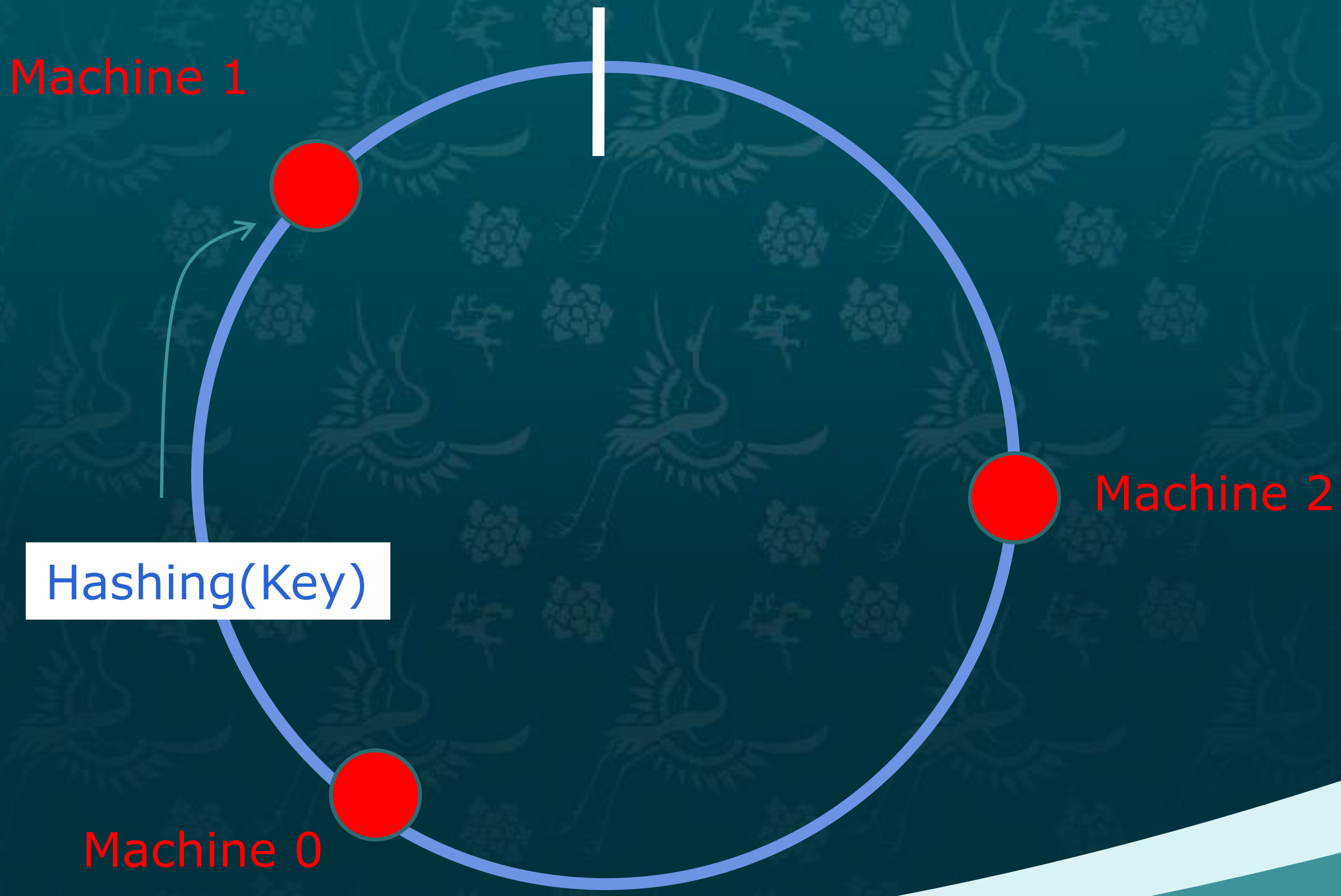
# Fundamentals of Consistent Hashing

- Basic Concept
- Example

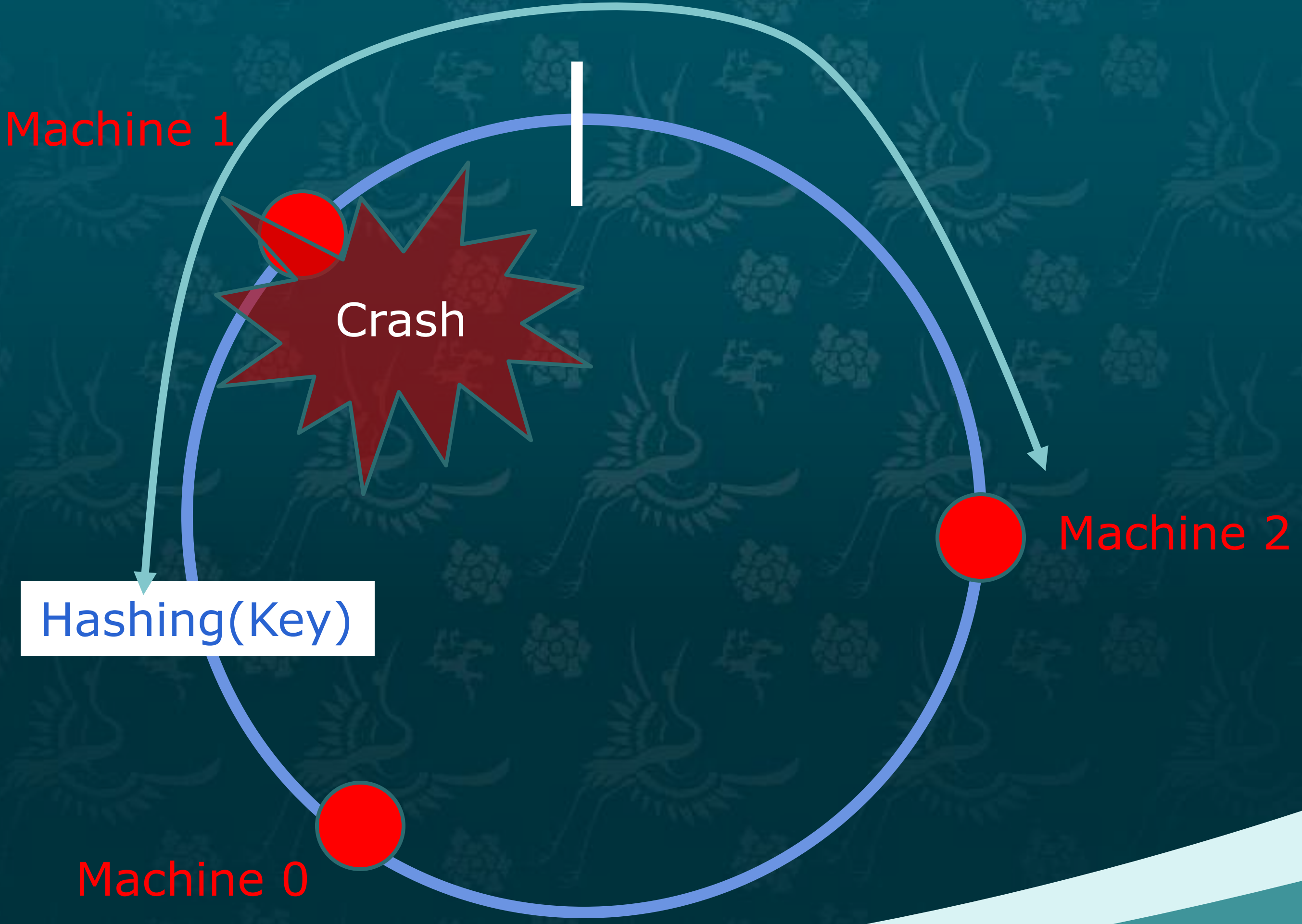
# Basic Concept

- Hashing machine number (j)
  - Hash Function(j)
- Distribute keys to near machine depending on the hash number
  - Clockwise of the key.
- Reallocate rate
  - 1/rn (j = machine number, r = replica count, n = machine count)
  - (j,0), (j,1)....(j,r-1)

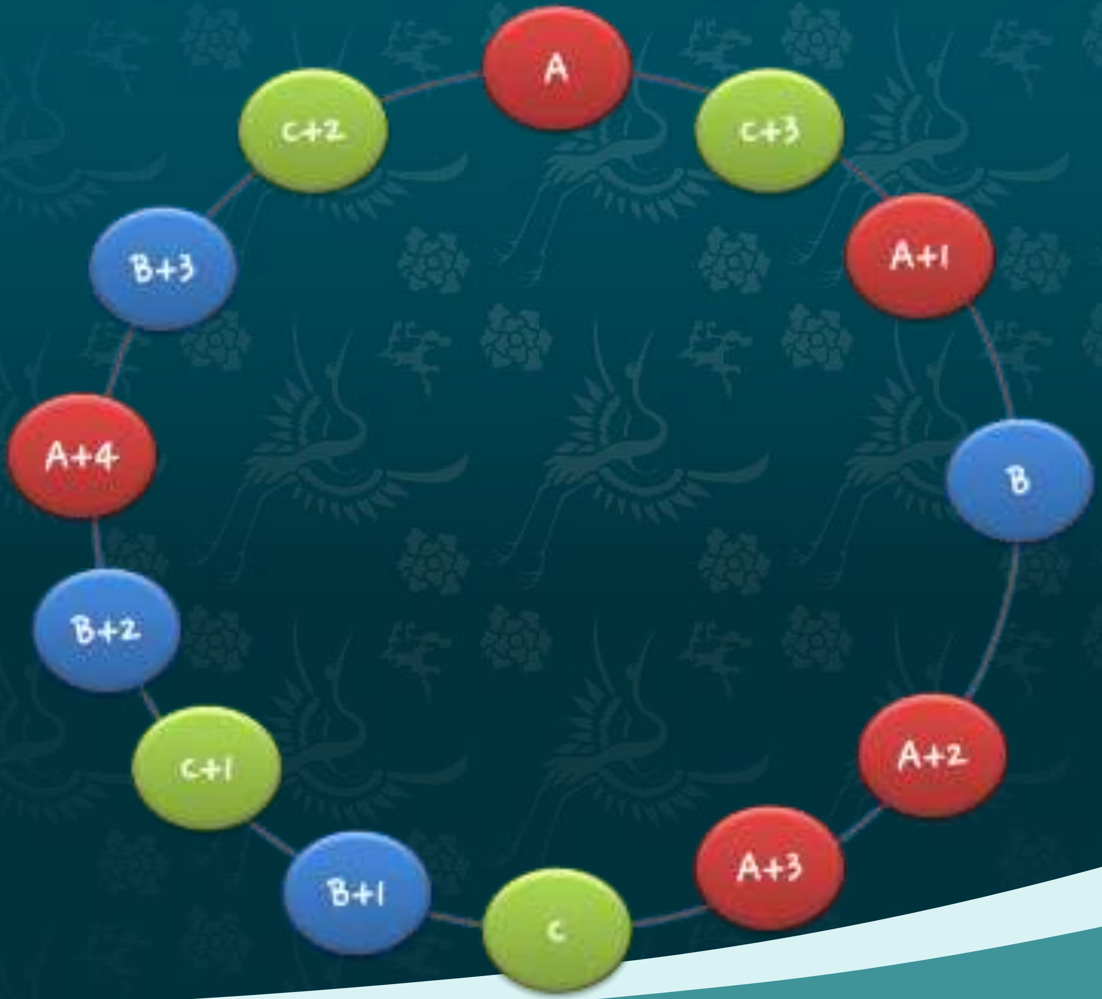
# Example-1



# Example-2



# Example-2



# Strength of Consistent Hashing

- Few keys should be re-allocated
- Easy Horizontal Scalability
  - Expand
  - Shrink
- Better Performance
  - Similar keys in near machine or same machine

# Reference

- http://charsyam.wordpress.com/2012/11/26/%EC%9E%85-%EA%B0%9C%EB%B0%9C-consistent-hashing-%EA%B3%BC-replication/

- http://charsyam.wordpress.com/2012/11/26/%EC%9E%85-%EA%B0%9C%EB%B0%9C-consistent-hashing-%EA%B3%BC-replication/