Consistent Hashing :

Dynamic Load distribution
and
Load balancing

BY: MONIKA SHAH

DATE: 23-4-2016

Outline

Motivation

Key Features

Research Applications

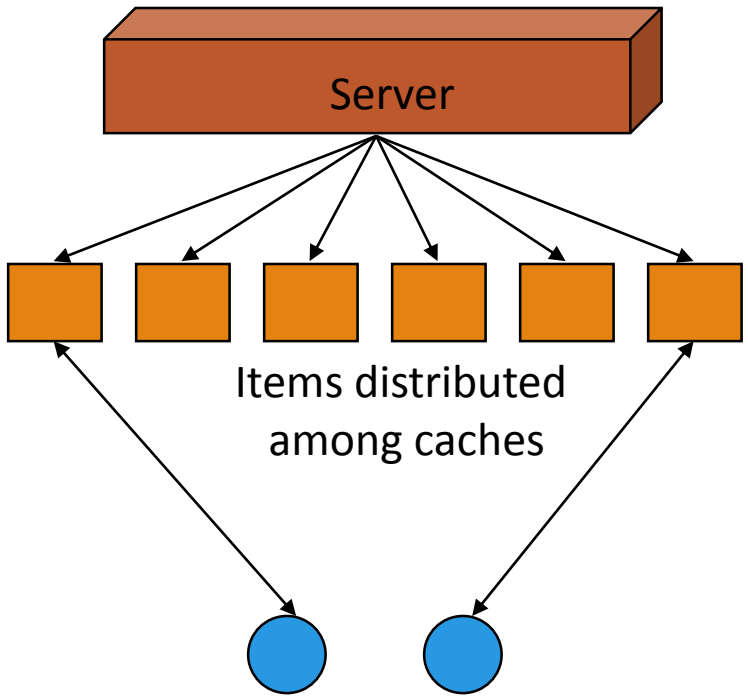
Introduction

Background

Challenges

Application

Motivation case study: Web caching



Known fact: caching for improving efficiency of data delivery over internet

Single cache – problems:

- Cache size limit
- Single Point Of Failure
- Swamped by user requests

Solution : Hashing based Distributed Web Cache servers

- Cache server ***s = Hash(URL)***
- User get content from cache server ***S***

Problems:

- ***What if node joins/ fails ?***
- ***What if different view?***

Solution : Consistent hashing

Key features

- Dynamic Load distribution
- Better Scalability
- Efficient Resource utilization
- Efficient Data retrieval
- Improving Performance
- Parition Tolerance

Research Applications

- Design for **Bigdata storage and efficient retrieval** using NoSQL Database over cloud
  - E.g.. Cassendra [2], Amazon's Dynamo[3]
- **Cloud storage mgt.:** Data placement strategy in Cloud computing[7]
- **Virtual Partition Consistent Hashing (VPCH) :** Balancing reducer load in map-reduce [8]
- ***UbiCrawler****: One of best* Distributed Web Crawler [5]
- **Distributed Real – time storage :** Fast, timely storage for IoT applications[6]
  - E.g. Transport monitoring for traffic management system

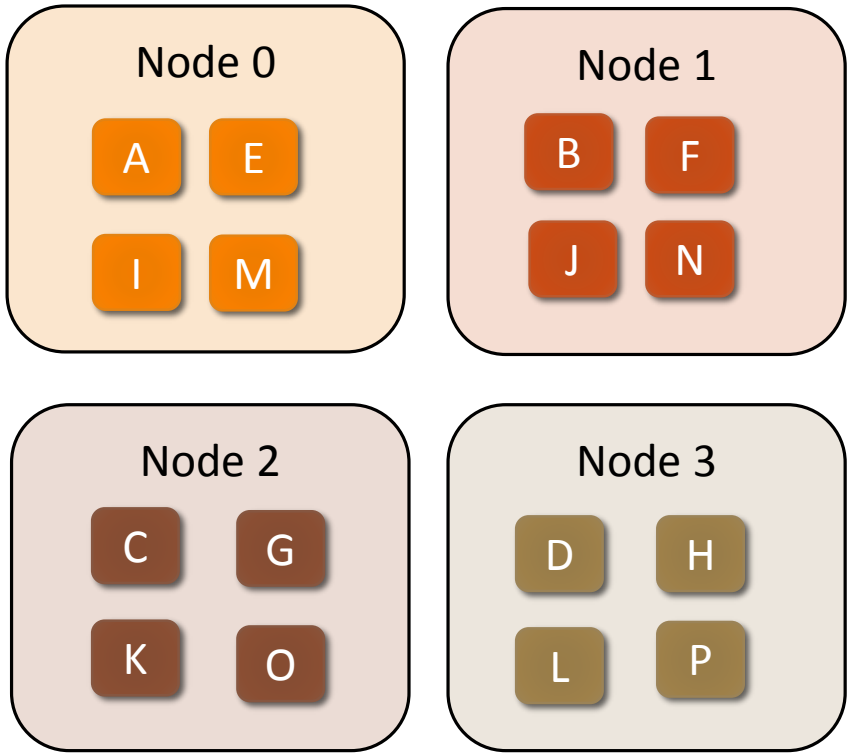
Other Applications:

Task: distribute items into buckets

- Data to memory locations
- Files to disks
- Tasks to processors
- Web pages to caches (our motivation)

Goal: balanced distribution

Background: Basic Hashing



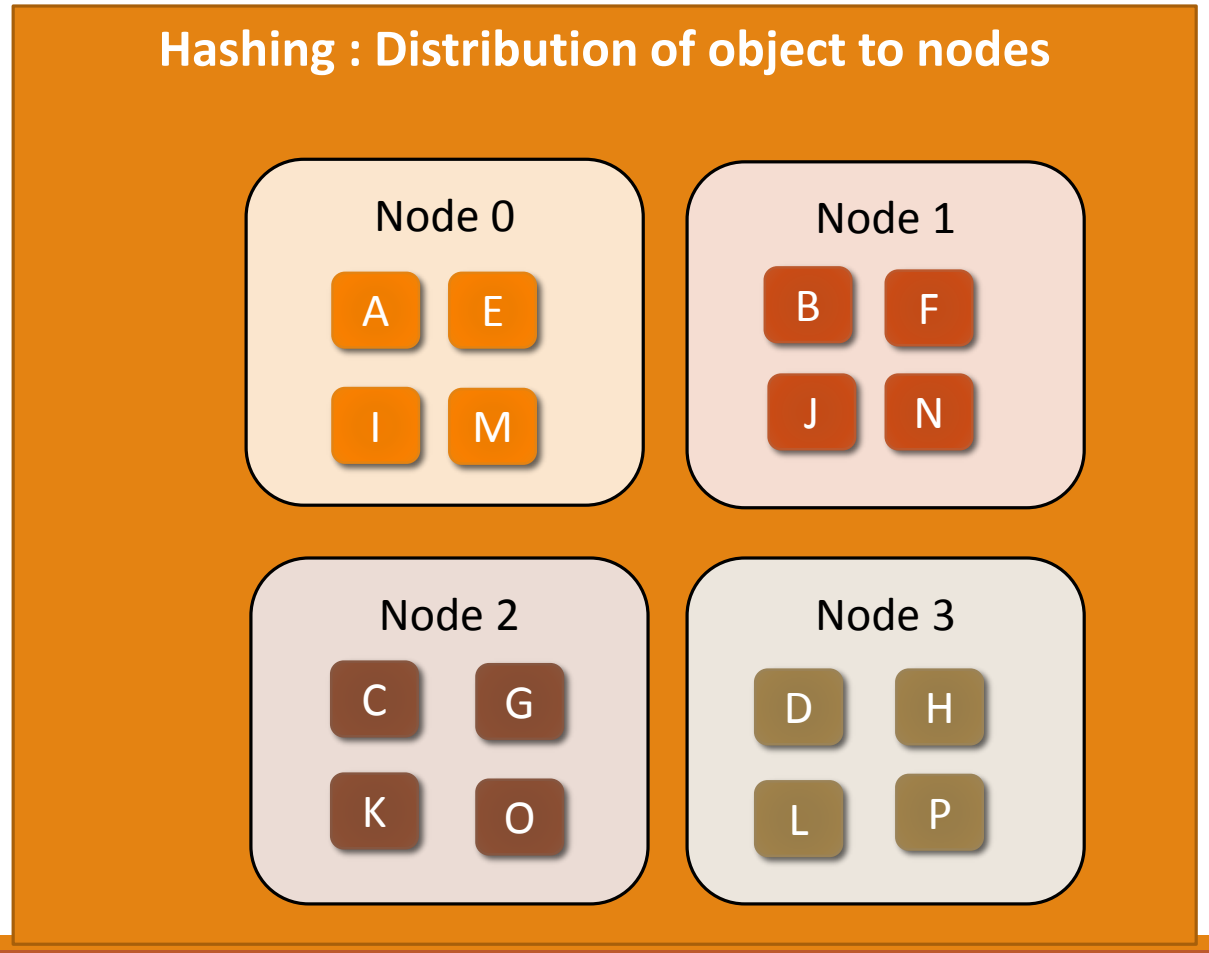
to distribute objects among cache servers

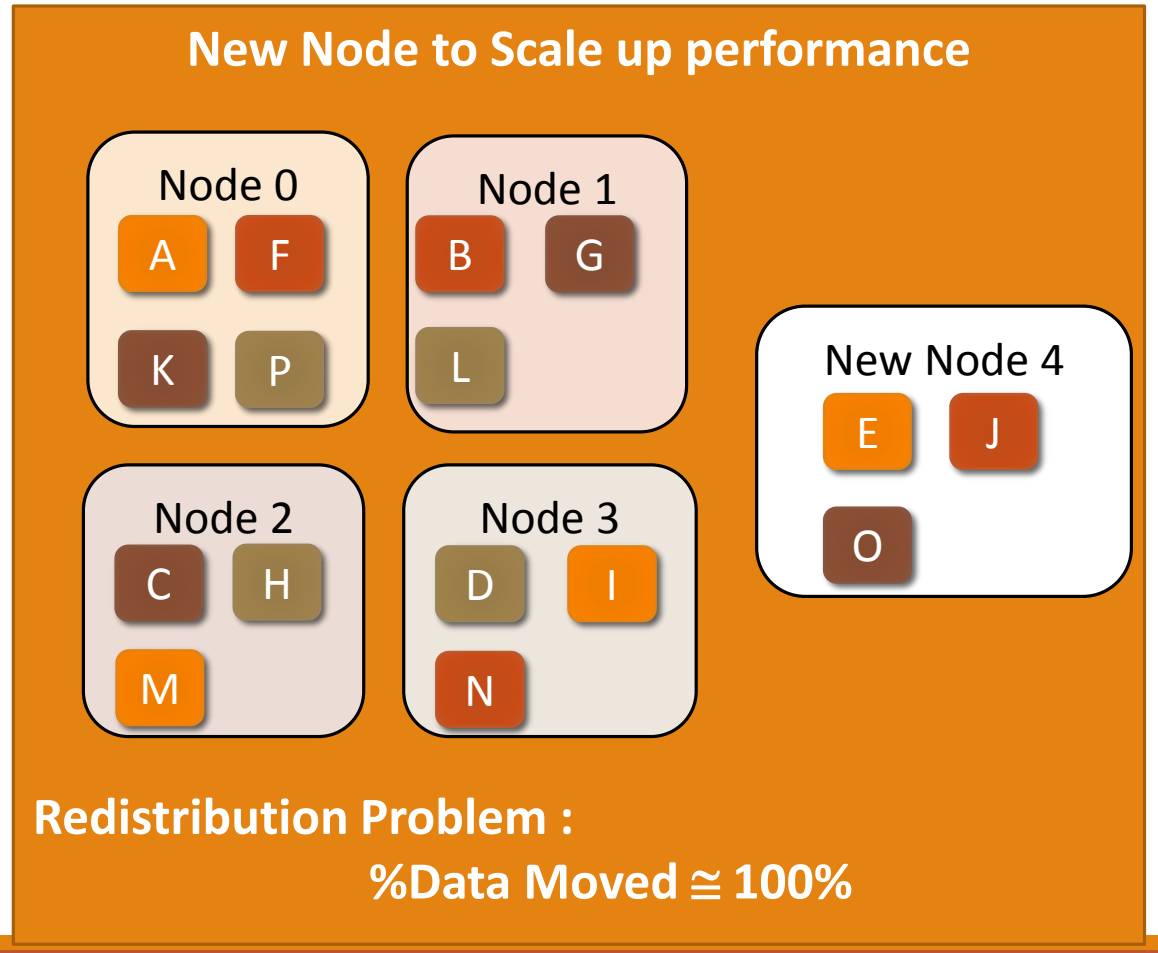
Goal of Hashing in Distributed System :

- **Balanced** distribution
- **Smoothness**: Little impact on buckets on
  - Adding a bucket (for scalability)
  - Remove a bucket (Node Failure)
- **Light Load :** on each Node
- **Spread** : small set of object that may hold an object

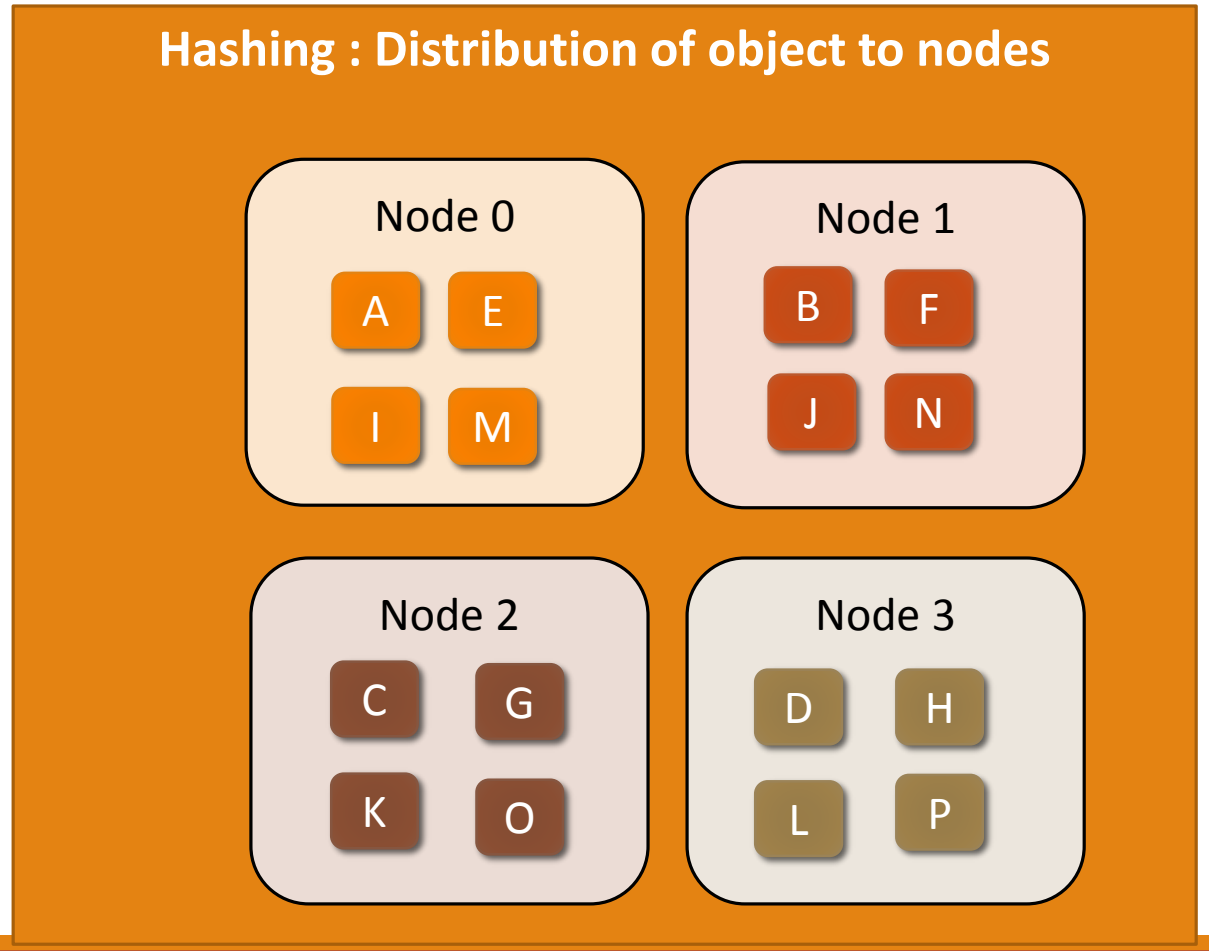
Simple Hash Function Location = Hash(Key) % # Nodes

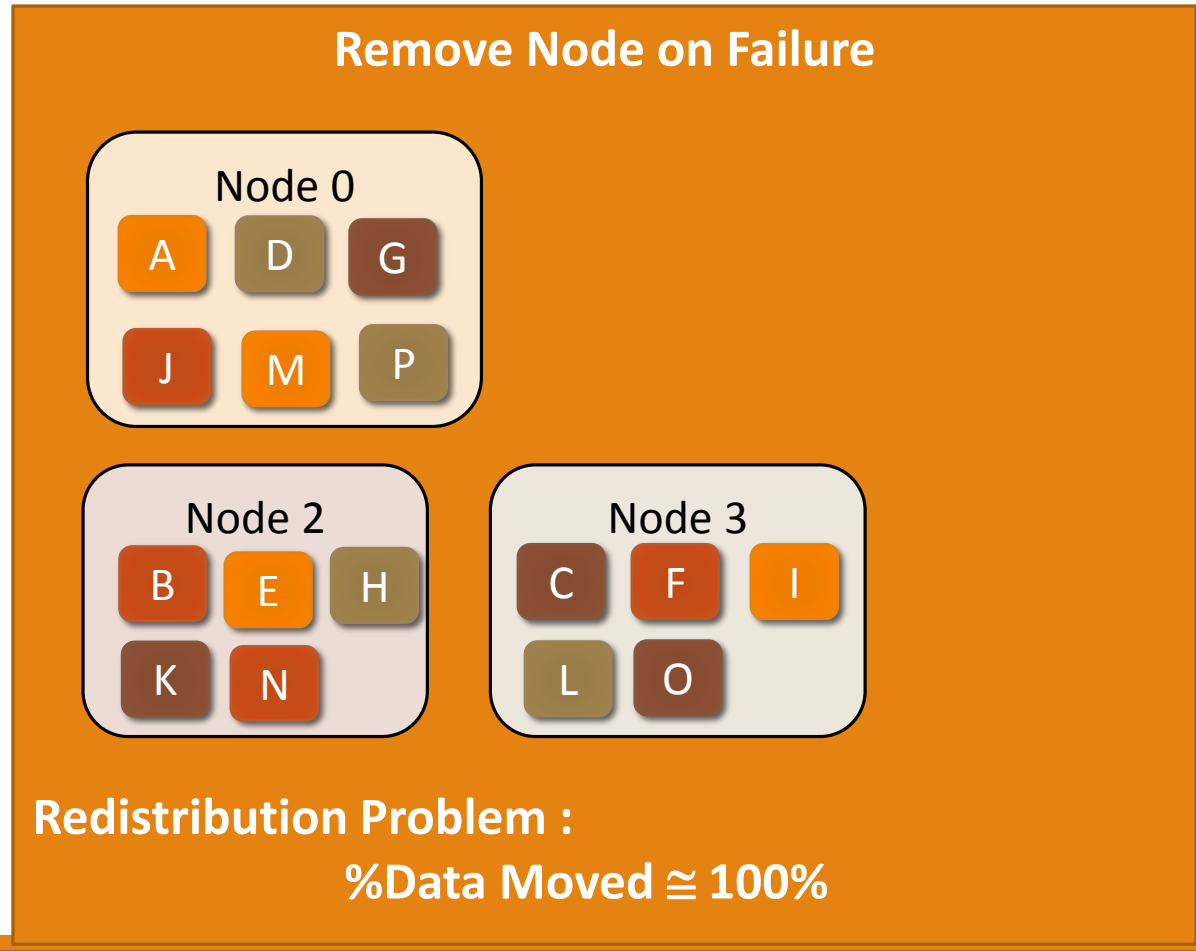
Hashing problem : Add/Remove node





Hashing problem : Add/Remove node

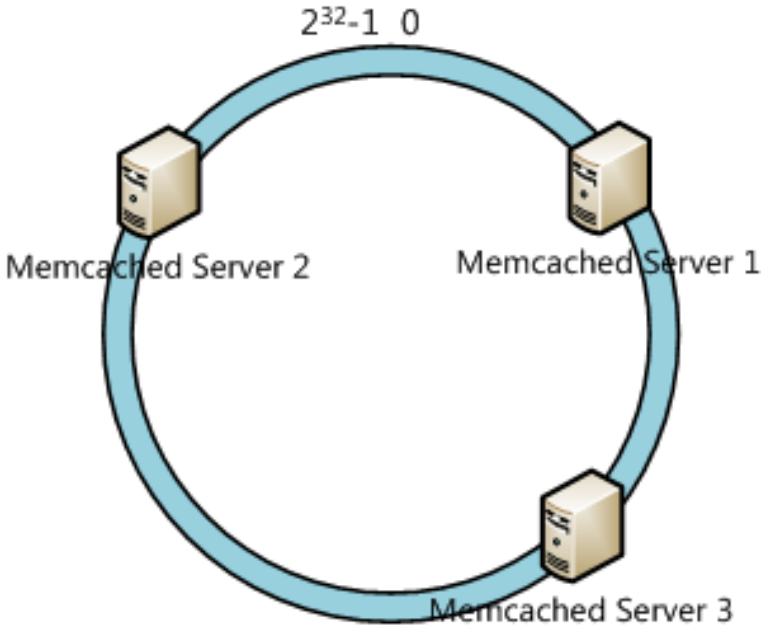




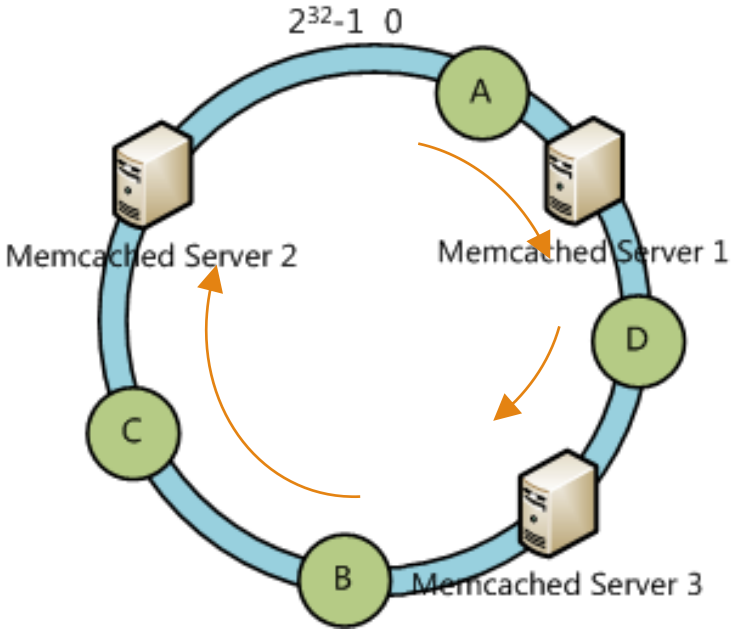
Consistent Hashing

Introduced by Karger et al.[1]

Consistent (same) hashing function for Cache (node) and data objects

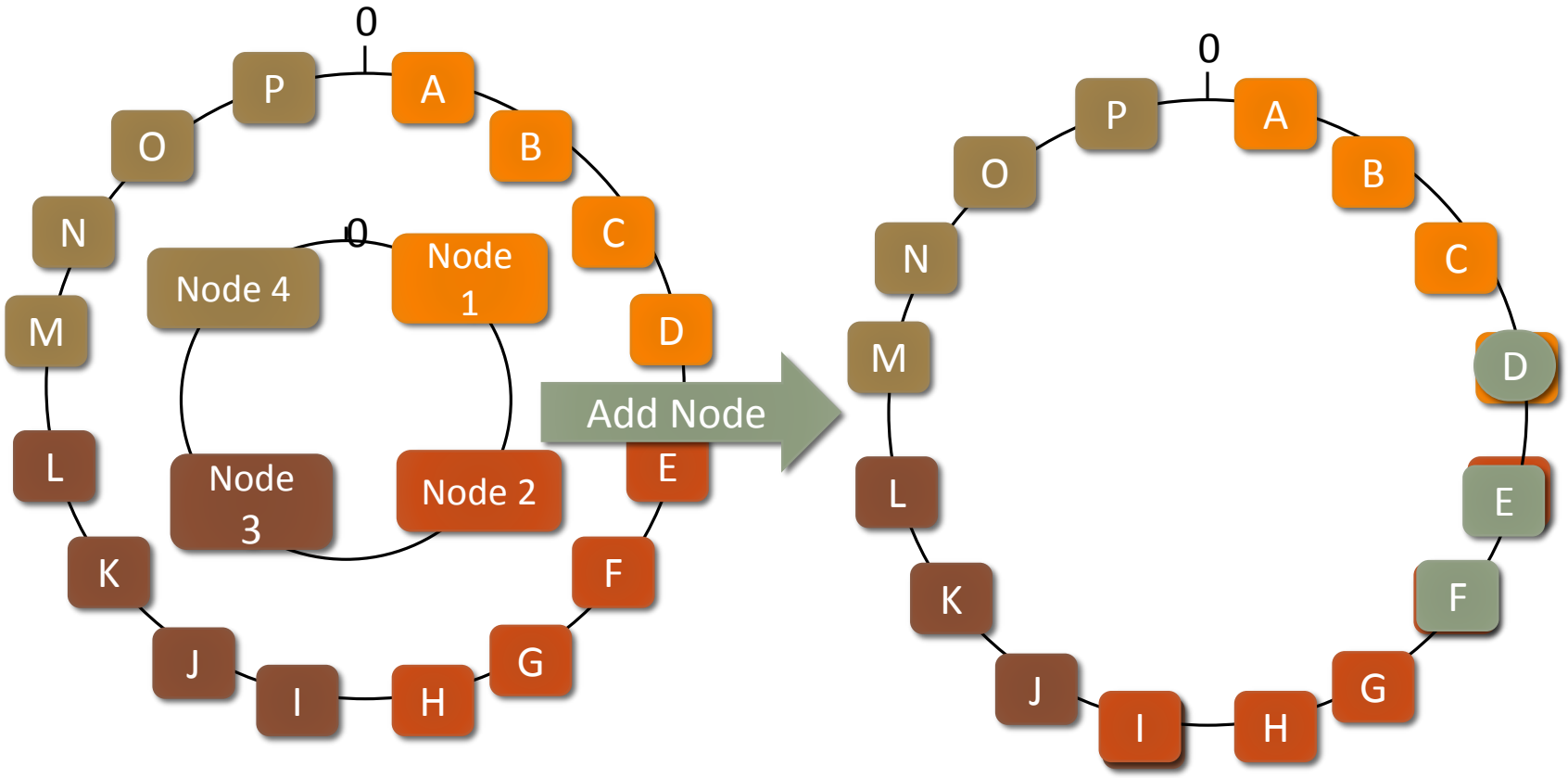


Hash server : i.e. Hash(IP)



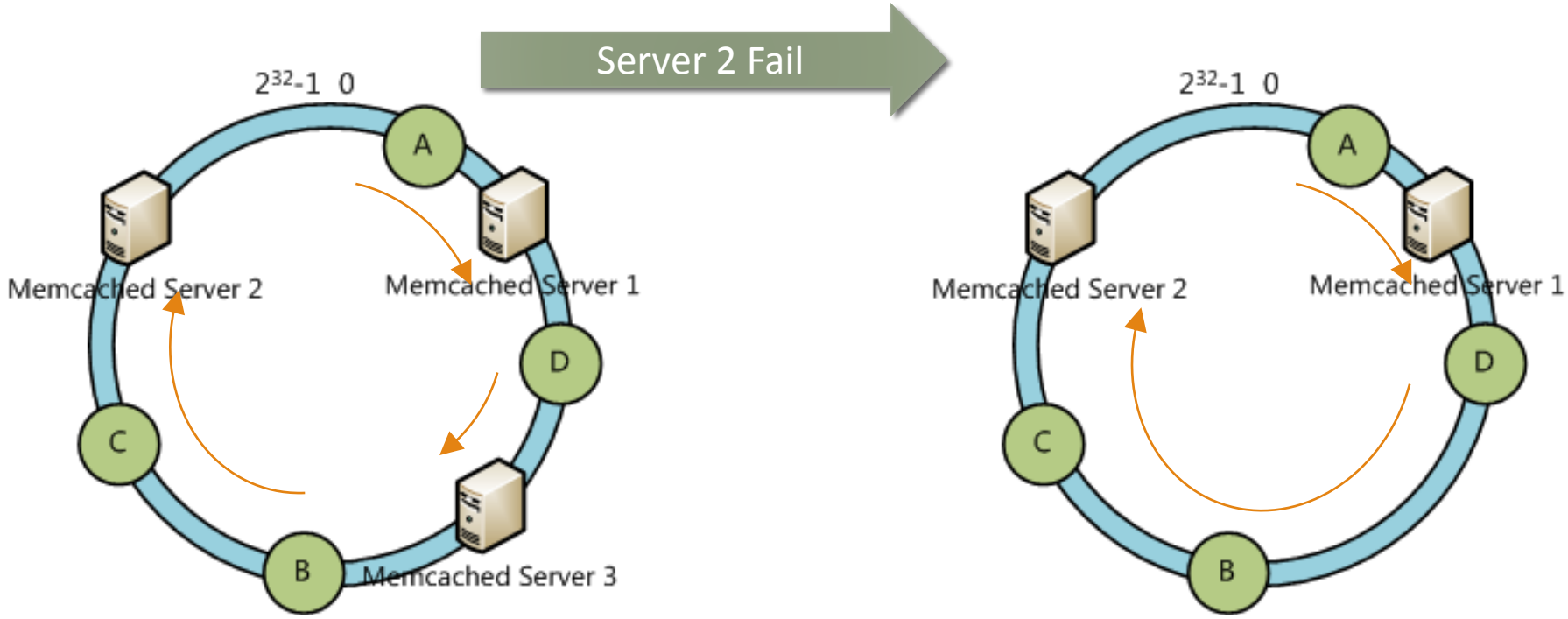
Hash objects: i.e. Hash(o) % servers

Redistribution on Joining of new node/Back up of failed node



Redistribution = % Data Moved = 100 * **1 / N** i.e. 1/5 * (16) = 3

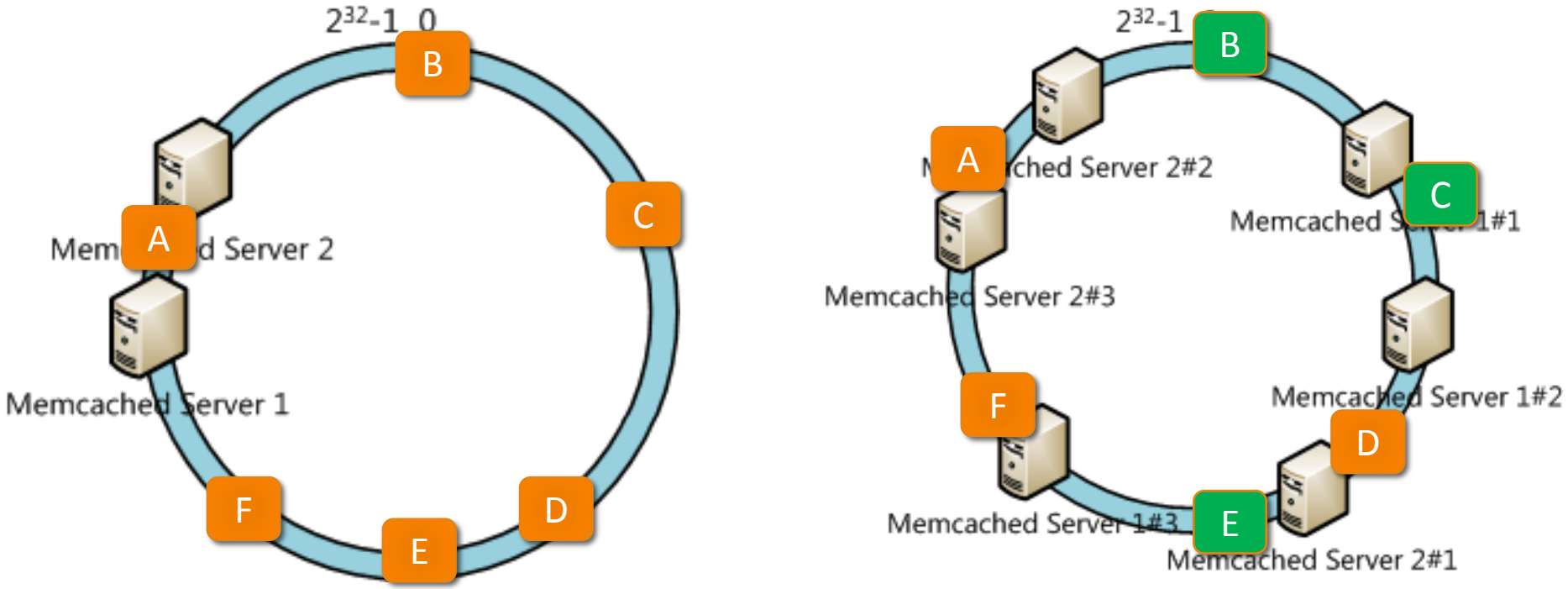
Consistent Hashing: Redistribution on a server down



Challenge : Balanced Distribution

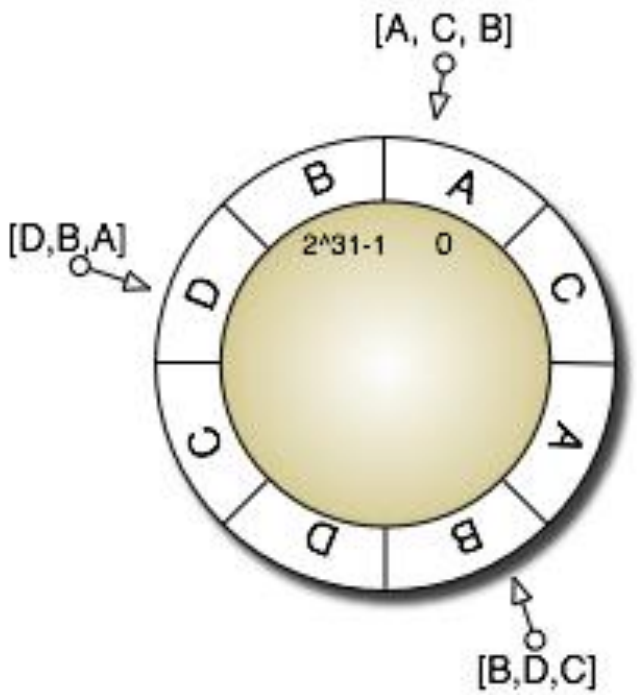
for efficient resource utilization and performance

Solution : Use Virtual Node for better balancing



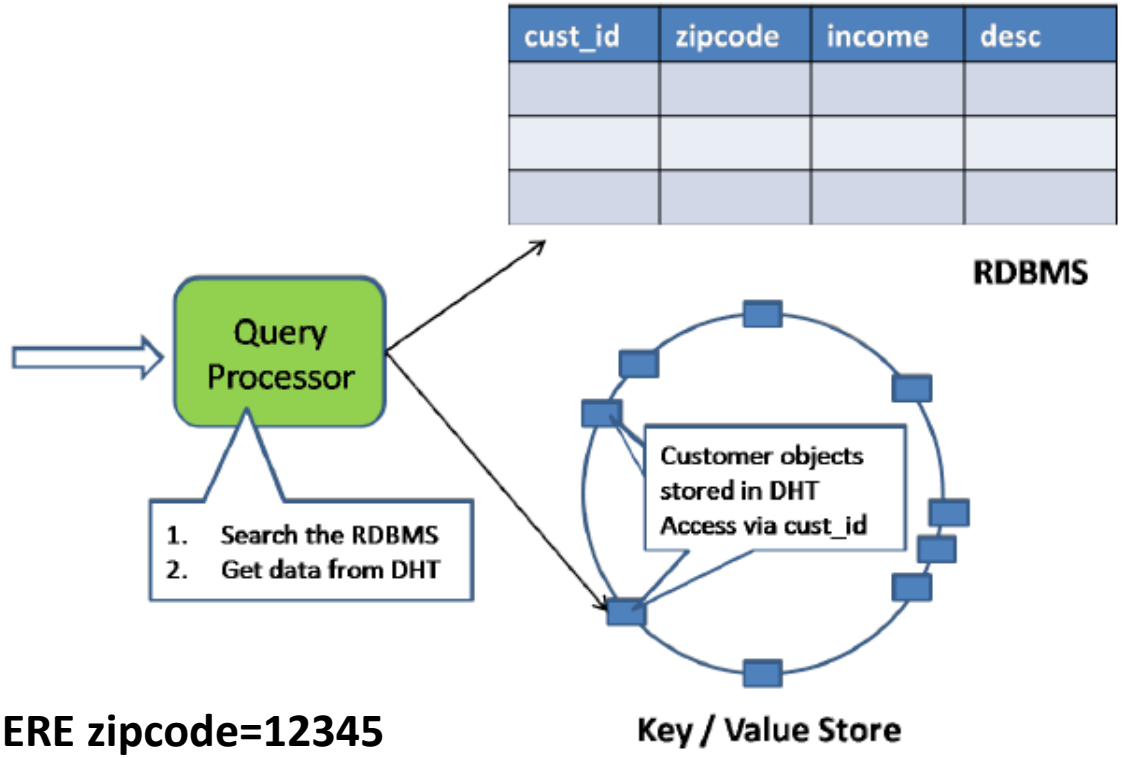
Challenge : Fault / Partition tolerance

Solution : Use Virtual Node for replica of object to nearer nodes



Challenge in designing Query model :
Data access by variety of predicates

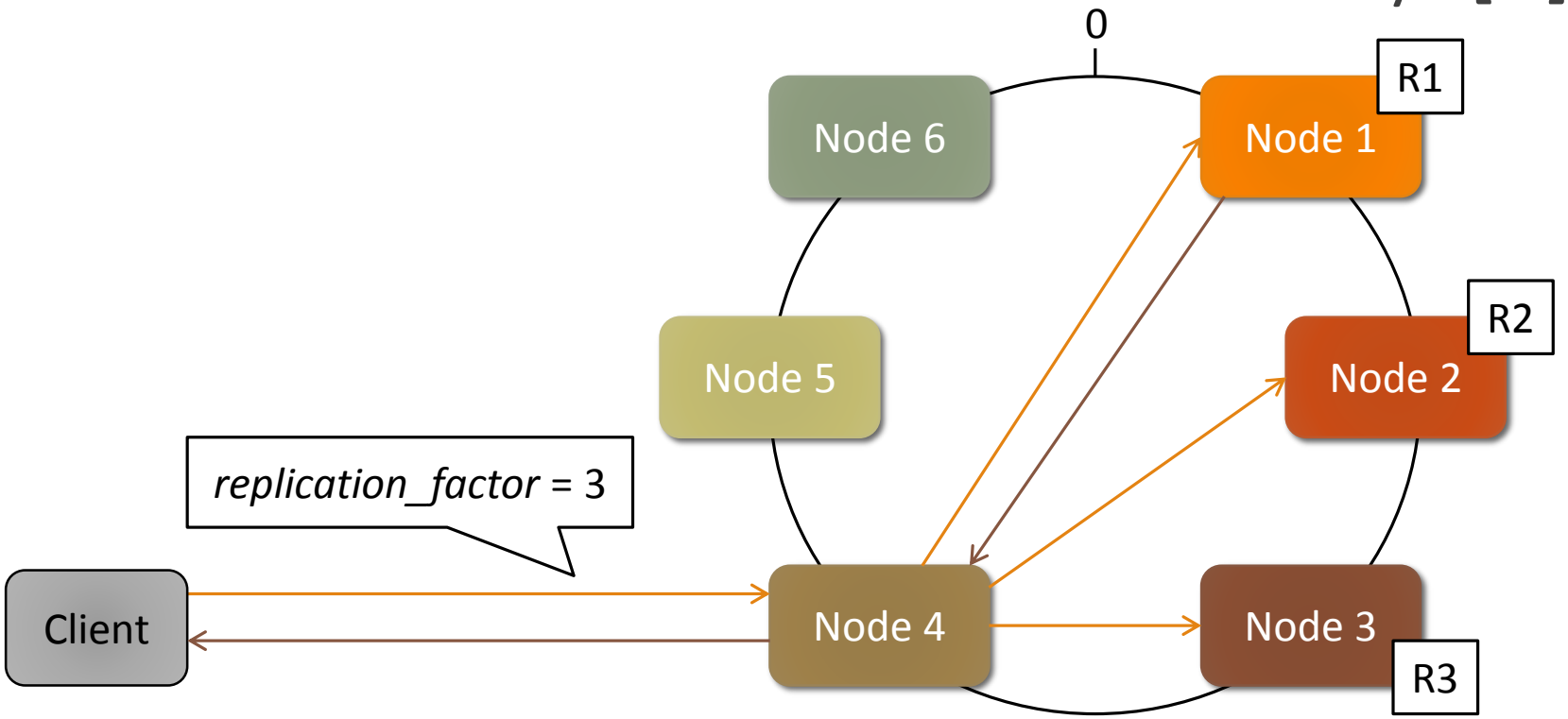
Solution : Use Key mapping for predicate attributes[4]



SELECT * FROM CUSTOMER WHERE zipcode=12345

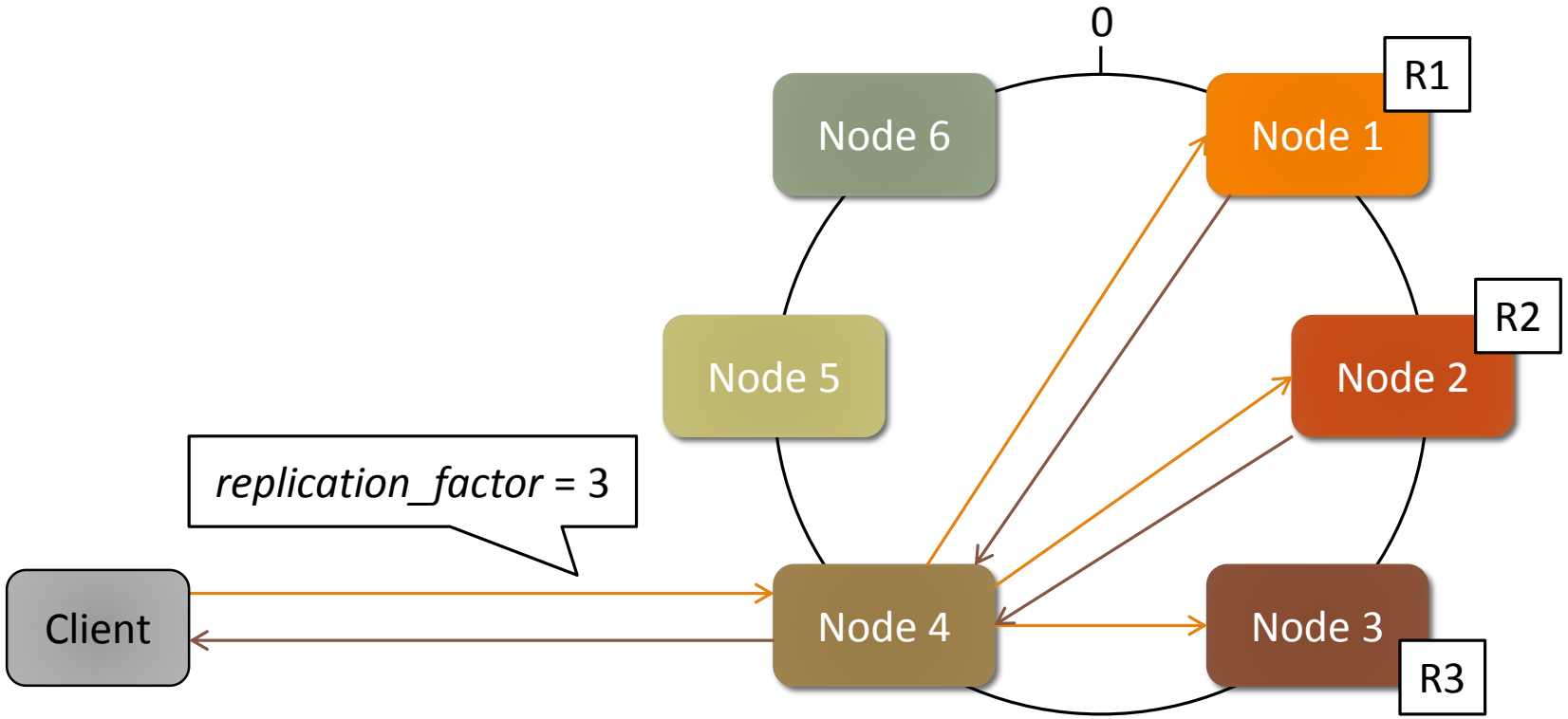
Challenge : Consistency

Cassandra solution: Tuneable consistency [3]



INSERT INTO table (column1, …) VALUES (value1, …) USING CONSISTENCY ONE

Challenge : Consistency
Cassandra solution: Tuneable consistency [3]



INSERT INTO table (column1, …) VALUES (value1, …) USING CONSISTENCY QUORUM

Challenges :

Deciding Hash function parameters (ie. URL, IP, website size, structure of site[5])

Parameters for Virtual node configuration for heterogeneous nodes (Power, memory capacity, disk capacity)

Configuring number of virtual nodes as per Energy Efficiency algorithm[9]

References

1. D. Karger, E. Lehman, T. Leighton, M. Levine, D. Lewin, R. Panigrahy, "Consistent Hashing and Random Trees: Distributed Caching Protocols for Relieving Hot Spots on the World Wide Web", In Proceedings of the *29th Annual ACM Symposium on Theory of Computing*, El Paso Texas, 1997, pp. 654-663.

2. G. DeCandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, and W. Vogels, "Dynamo: Amazon's Highly Available Key-Value Store," in *Symposium on Operating System Principles*, vol. 7, 2007, pp. 205–220.

3. A. Lakshman and P. Malik, "Cassandra: A Decentralized Structured Storage System," *ACM SIGOPS Operating Systems Review*, vol. 44, no. 2, pp. 35–40, 2010.

4. A. Lakshman and P. Malik, "Cassandra: A Decentralized Structured Storage System," *ACM SIGOPS Operating Systems Review*, vol. 44, no. 2, pp. 35–40, 2010.

5. Mitra Nasri1, Mohsen Sharifi2, "Load Balancing using Consistent Hashing: a Real Challenge for Large Scale Distributed Web Crawlers", In Proceeding of the International Conference on Advanced Information Networking and Applications Workshops, 2009, IEEE DOI 10.1109pp. 715-720

6. Yanhua Sun, Jun Fang, Yanbo Han, "A Distributed Real-time Storage Method for Stream Data", In Proceeding of 10th Web Information System and Application Conference, 2013, IEEE DOI 10.1109, pp. 314-317

7. Qiang Li, Kun Wang, Nigel Linge, "A data placement strategy based on clustering and consistent hashing algorithm in Cloud Computing", In proceeding of 9th International Conference on Communications and Networking in China (CHINACOM), 2015, IEEE DOI 10.1109, pp. 69-72

8. Qi Liu*, Weidong Cai, Jian Shen, Baowei Wang, Nigel Linge, "VPCH: A Consistent Hashing Algorithm for Better Load Balancing in a Hadoop Environment", In proceeding of Third International Conference on Advanced Cloud and Big Data, 2015, IEEE DOI 10.1109, pp. 69-82

9. Frezewd Lemma, Thomas Knauth, Christof Fetzer, "PowerCass: Energy Efficient, Consistent Hashing Based Storage For Micro Clouds Based Infrastructure", In proceeding of IEEE International Conference on Cloud Computing,, 2014, IEEE DOI 10.1109, pp. 48-54

Thank You