Building Faster, Elastic, and Durable Large-scale Data Store with Consistent Hashing

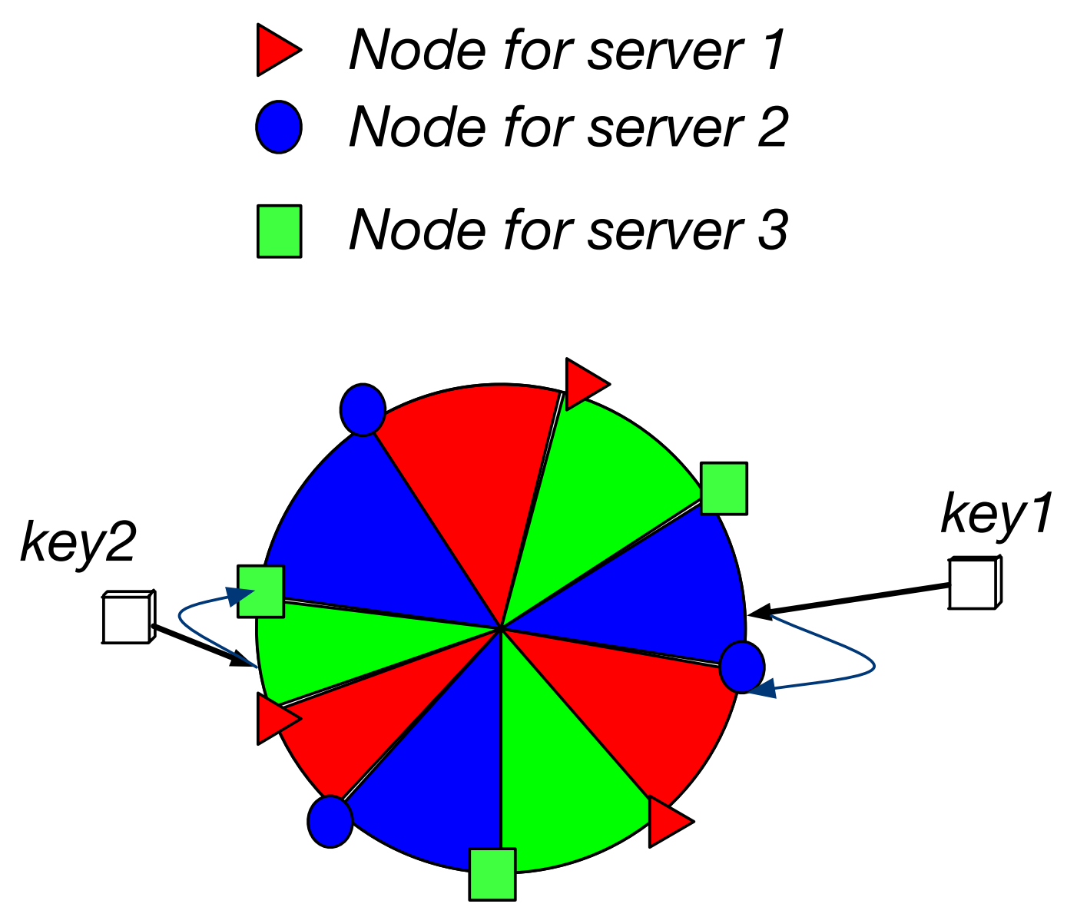
Wei Xie and Yong Chen
Department of Computer Science, Texas Tech University

Abstract

- Many large-scale data store uses the consistent hashing algorithm or its variants for better scalability and manageability, e.g. Dynamo, Cassandra, Ceph, Sheepdog.
- Lacking support for heterogeneous storage devices and elastic storage.
- Propose of a ***two-mode consistent hashing*** algorithm that better support heterogeneous storage devices to offer both **performance improvement** and **balanced capacity utilization**.
- Propose of an *elastic consistent hashing* algorithm to offer **agile cluster resizing** and **selective data re-integration**.

Consistent Hashing

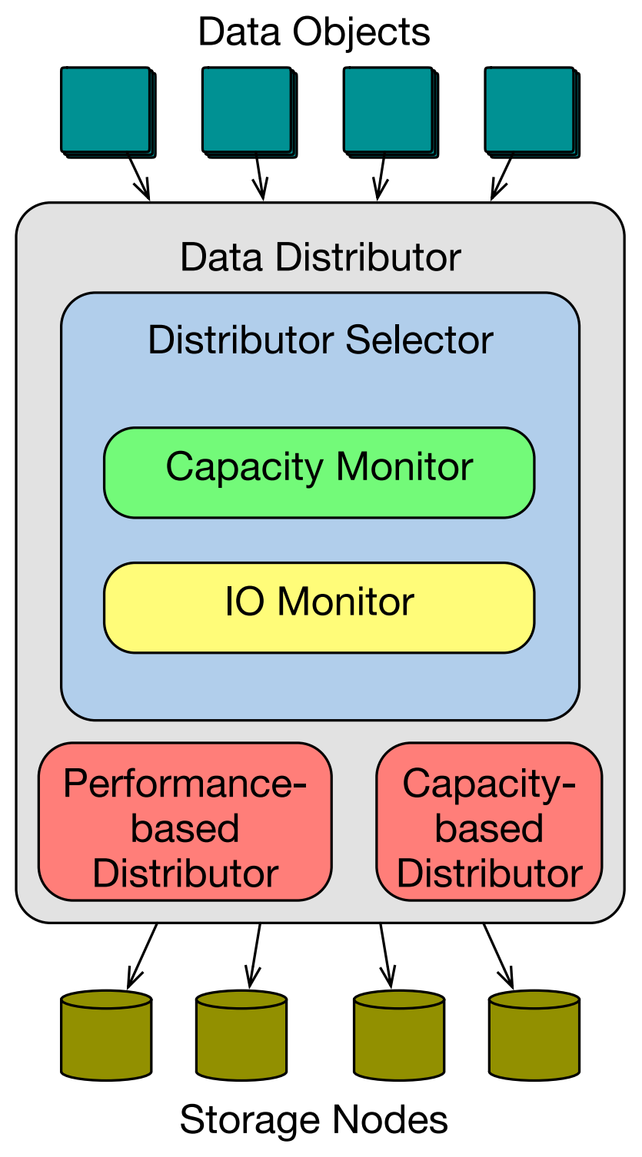
- Initially used for load balance in web caching
- Each server generates one or multiple nodes on a hash ring
- A key (data) generates a node on the ring as well and matches to the next server node in the clockwise direction



Research Problem and Existing Solutions

- Support heterogeneous storage
  - Flash-based SSD and HDD co-exist in many large-scale storage system
  - SSDs offer better performance but have small capacity
  - HDDs have much more abundant capacity in most large-scale systems
  - Consistent hashing only puts weights on storage servers according to their capacity, which could underutilize the SSDs' performance
  - Existing heterogeneous storage systems are managed via a caching layer or tiered storage solution, which requires an extra layer to manage heterogeneous devices
- Support elastic storage
  - Many large-scale storage systems resize cluster according to workload demand to save power consumption
  - Need an elastic data layout that a full data copy stored on a small set of servers
  - Resizing may incur excessive data migration that degrades performance
  - Existing study like SpringFS only works on HDFS-like distributed file systems

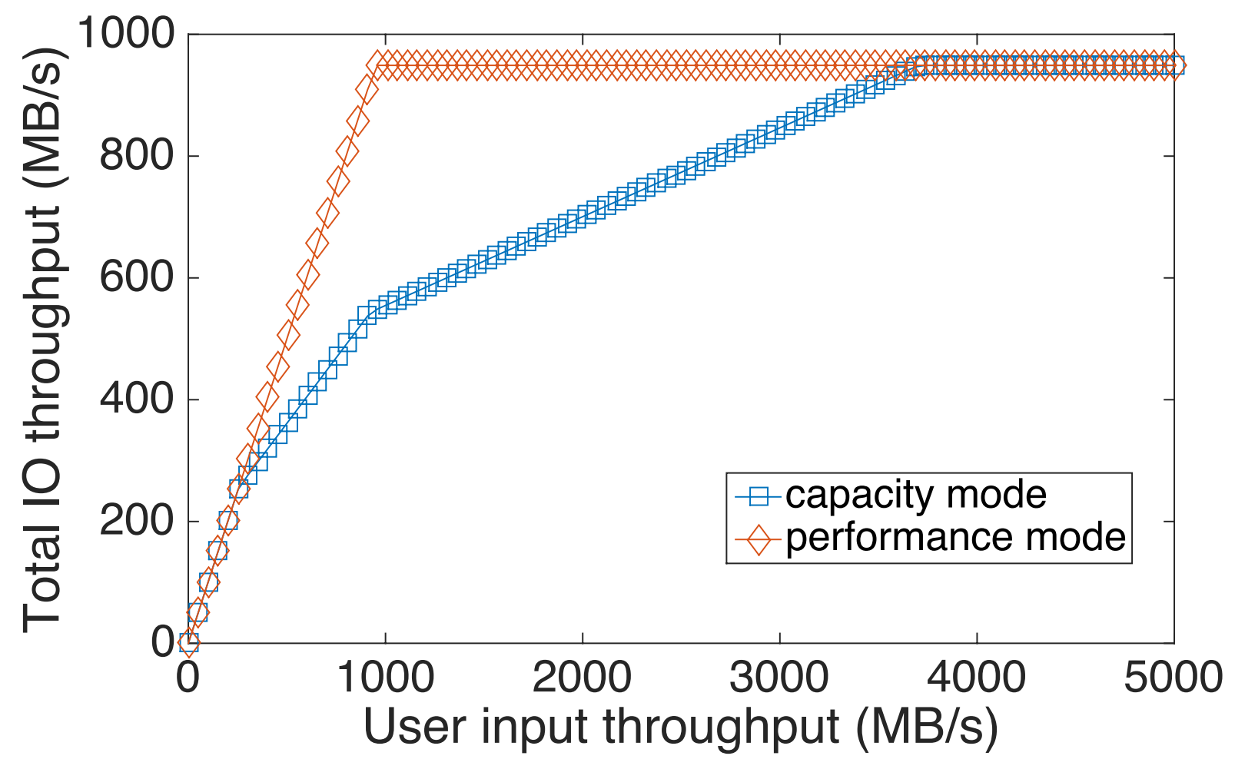
Two-Mode Consistent Hashing



- Performance mode: weight of nodes proportional to device throughput
- Capacity mode: weight of nodes proportional to device capacity

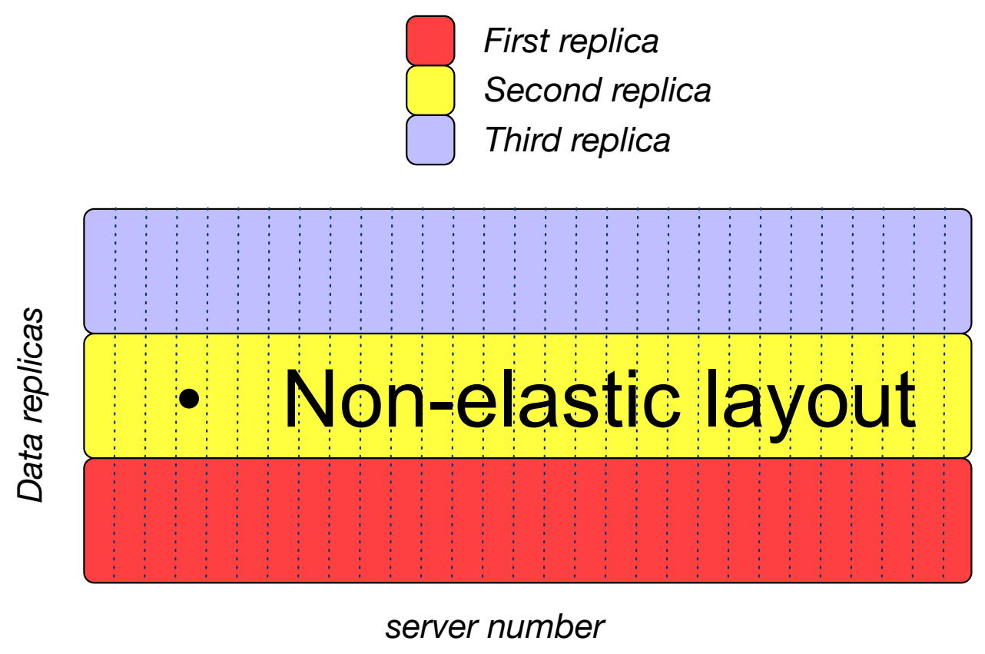
Node name	Capacity (GB)	Throughput (MB/s)	Number of VNode (capacity mode)	Number of VNode (performance mode)
Node S	250	70	4	10
Node H	500	350	8	2

- Capacity monitor: when variance of capacity exceeds threshold, switch to capacity mode for load balance
- IO monitor: when IO load is low, switch to capacity mode

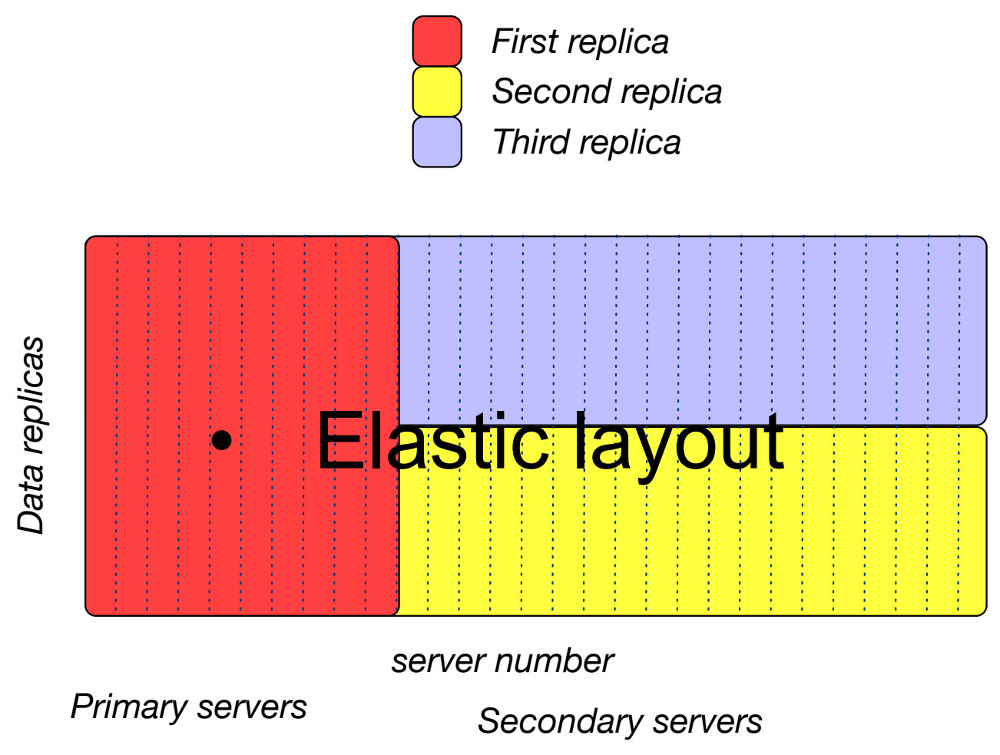


- Findings:
  1. Performance offer significant improvement on write performance
  2. Two-mode does not increase data distribution time significantly (worst case is to use two distributors to locate data)
  3. Mode transition overhead can be mitigated by background data migration

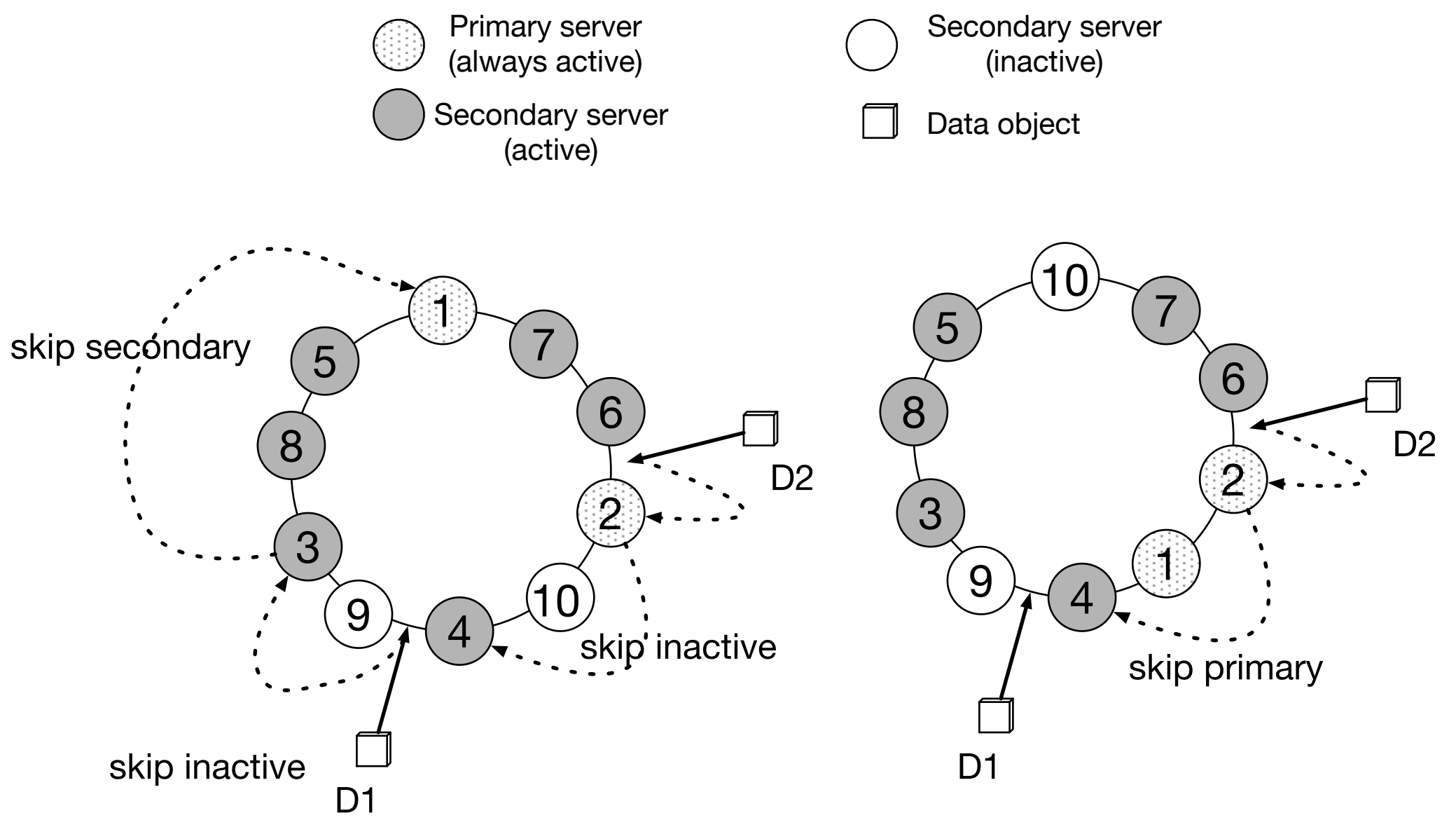
Elastic Consistent Hashing



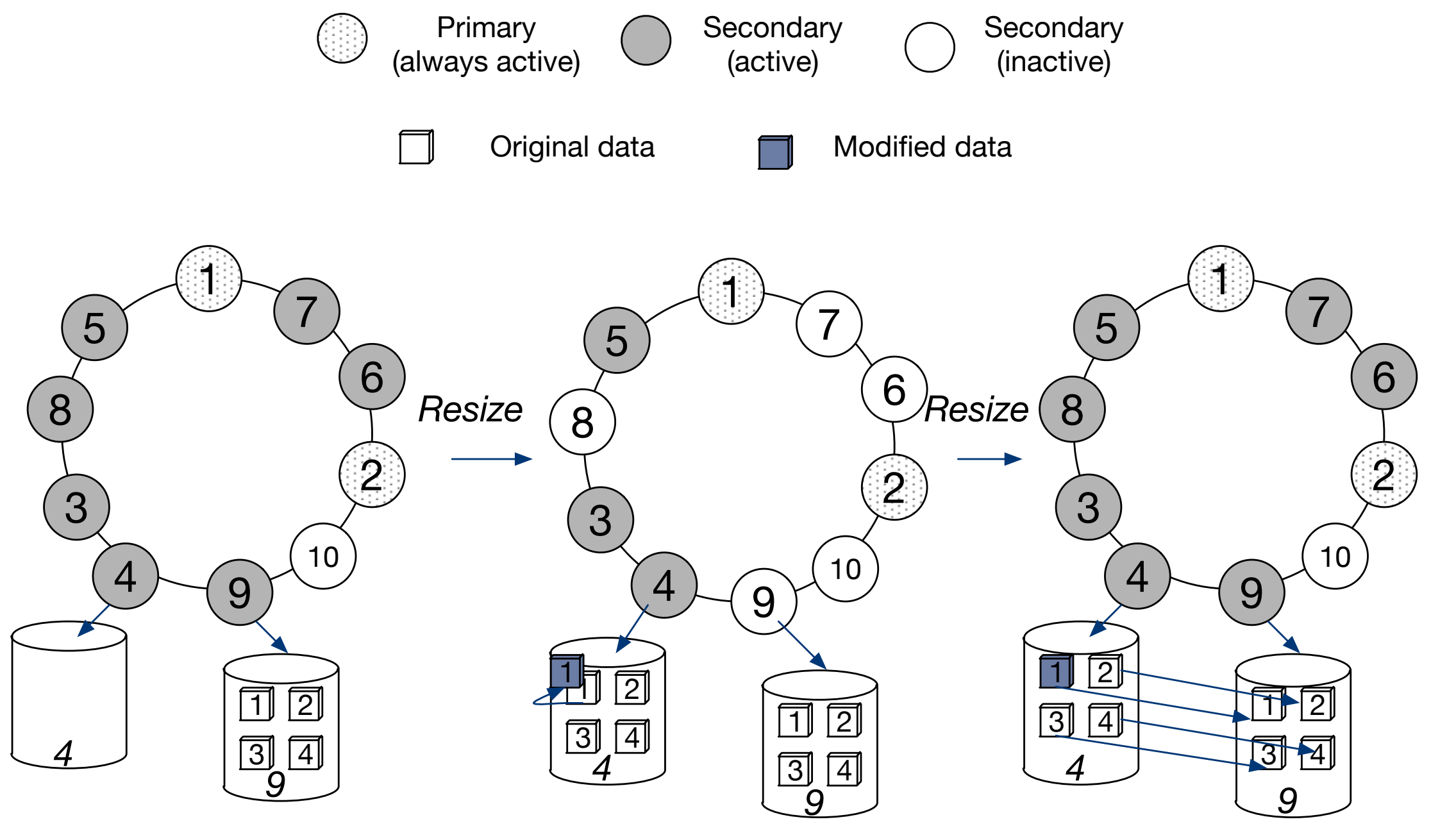
- Non-elastic layout

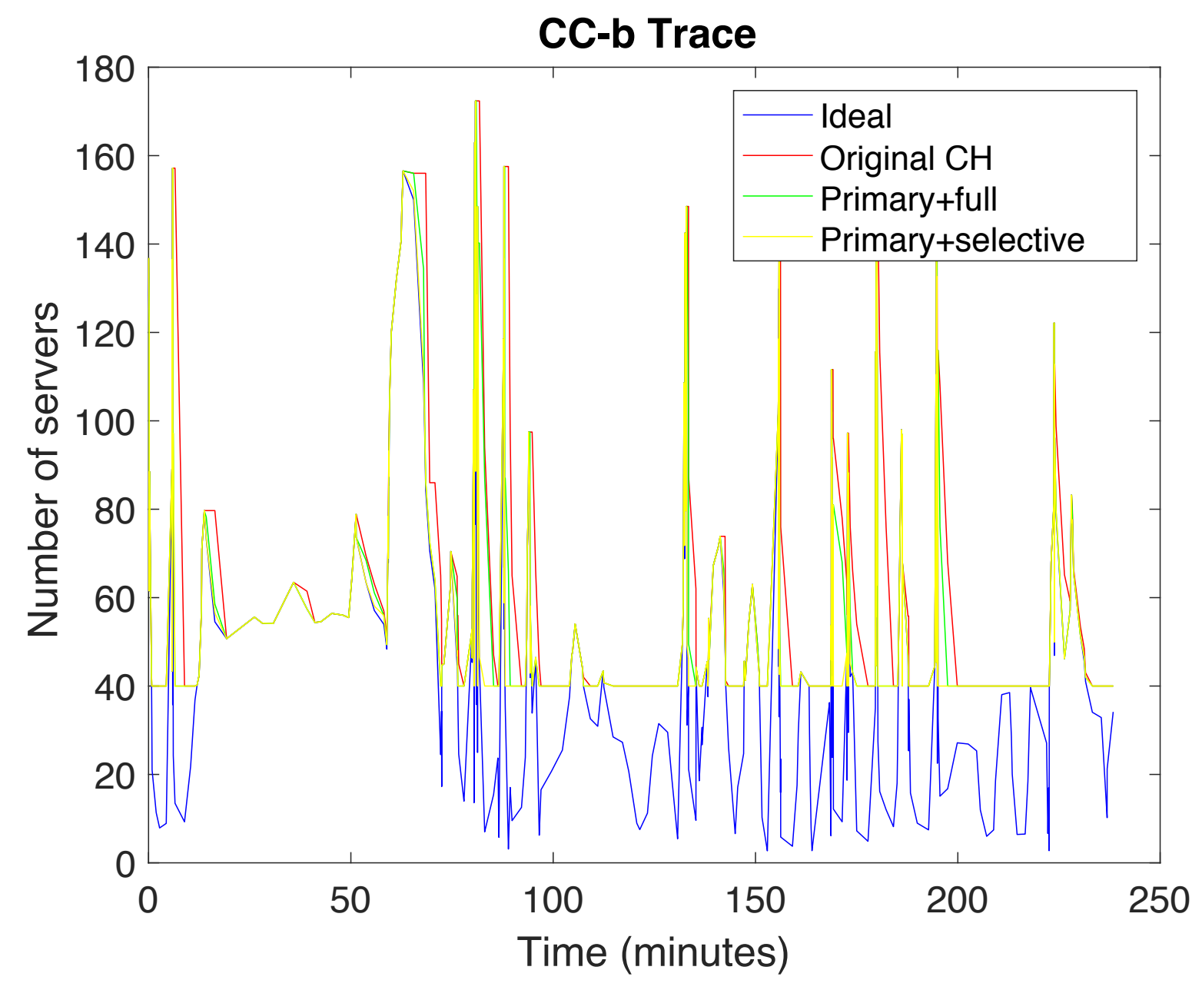


- Elastic layout


Primary server (always active)
Secondary server (active)
Secondary server (inactive)
Data object
skip secondary
skip inactive
skip inactive
skip primary
D1
D2
D1
D2
1
2
3
4
5
6
7
8
9
10

- An elastic layout ensures that the first copy is always available if the primary servers are active
- Primary server design to achieve elastic data layout





- Selective data re-integration
  - When sizing up, only migrate those data that have been modified
  - Each resize is associated with a version
  - The modified data in each version are recorded
- Findings:
  - Elastic layout avoids delay of size-down
  - Selective re-integration avoids extra migration workload that requires extra node to turn on, thus better machine hour saving
  - Saves 8% to 12% machine hours compared to resizing via original CH

Summary

- Consistent hashing algorithm is a promising solution for large-scale data stores
- We propose two variants of consistent hashing to achieve a high performance and power-efficient distributed data store

Acknowledgements

The research studies are supported in part by the National Science Foundation under grant CNS-1162488, CNS-1338078, and CCF-1409946.

31st IEEE International Parallel &
Distributed Processing Symposium , May 29 – June 2, 2017
Buena Vista Palace Hotel, Orlando, Florida USA