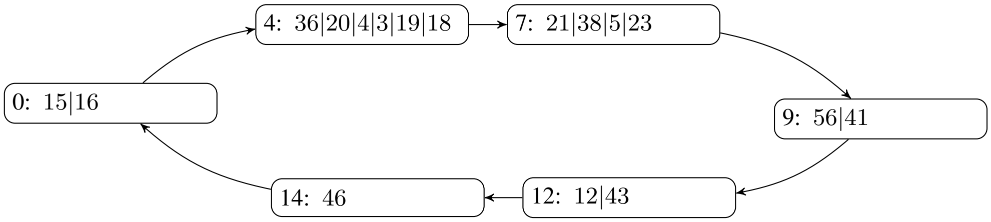
Exercises Big Data Management

Database Technologies and Information Management (DTIM) group
Universitat Politècnica de Catalunya (BarcelonaTech), Barcelona
March 25, 2025

1 Key-Value Stores

Problems

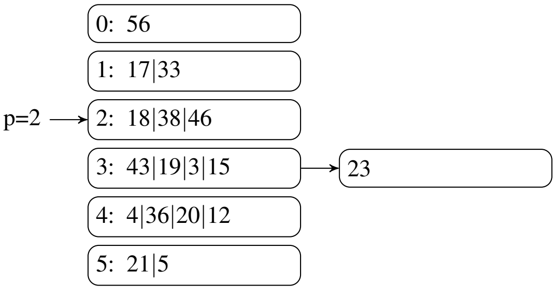
1. Let's assume we have a *Consistent hash* with $D = 16$, and the hash function is simply the module of the IP address or the key, and suppose the current state of the consistent hash is (position_in_the_ring:key|key|...):



   a) What happens when we insert objects 30 and 58? Draw the result.

   b) What happens in the structure when we register a new server with IP address 37? Draw the result.

2. Let's suppose we have a *Linear Hash* and the hash function is simply the module of the key, the capacity of a bucket is only four entries, and current state of the linear hash is (bucketID: key|key|...):



   a) What happens in the structure when we insert keys 14, 27, 37, and 44? Draw the result.

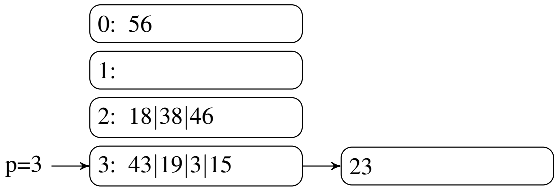
3. Let's suppose that, we have an *LSM Tree* that reached the threshold to consider the `MemStore` is full, and it contains four entries with format $[key, value, timestamp]$ needing ten characters each attribute (i.e., 30 overall). The content of the different structures is:

   - MemStore: $[1, v, t50], [15, v, t49], [17, v, t47], [29, v, t48]$
   - Commit Log: $[17, v, t47], [29, v, t48], [15, v, t49], [1, v, t50]$
   - SSTable$_{\text{Data}}$: $[13, v, t23], [25, v, t17], [35, v, t40], [59, v, t38]$
   - SSTable$_{\text{Index}}$: $[13, 0], [25, 30], [35, 60], [59, 90]$

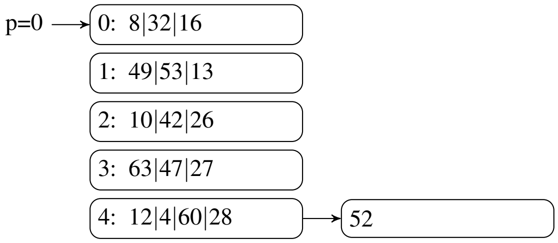
   Assuming that the minimum size of an `SSTable` is 120 characters and on having two `SSTables` a minor compactation is automatically triggered, explicit the content of all structures once the compactation is done.

   - MemStore:
   - Commit Log:
   - SSTable$_{\text{Data}}$:
   - SSTable$_{\text{Index}}$:

4. Briefly explain what is wrong in this linear hash structure, or if you think it is right, explicitly say so.



5. Given the **linear hash** underneath with $f(x) = x$ (i.e., we directly apply the module to the keys), and a capacity of four keys per bucket, indicate if it corresponds to a state **valid or not**. If valid, give a possible insertion order leading to it. If not, clearly explain why.



6. Suppose you have a hash function whose range has size 100 (i.e., D=100), and a Consistent Hash structure with 5 machines (M1...5) whose identifiers map to values $h(M1) = 0$, $h(M2) = 20$, $h(M3) = 40$, $h(M4) = 60$, $h(M5) = 80$. What happens if you have an object mapped to value $h(O) = 90$?

7. Given an empty **Consistent Hash** with $h(x) = x\%32$ (i.e., we directly take module 32 to both the keys and the bucket IDs), and unlimited capacity in each bucket, consider you have a cluster of four machines with IDs 19, 22, 75, 92, and draw the result of inserting the following keys in the given order: 12, 4, 10, 49, 42, 60, 63, 53, 47, 27, 26, 28, 13, 52.

8. Suppose you implement a system to store images in hundreds of machines with thousands of users using HBase with a single column-family. These images taken at time VT belong to a person P who tags each with a single subject S (e.g., family, friends, etc.) and are concurrently uploaded into the system at time TT in personal batches containing multiple pictures of different subjects taken at different times. Each person can then retrieve all his/her pictures of one single subject that were taken after a given time. Precisely define the key you would use (which cannot be a hash) if you exclusively prioritize (i.e., do not consider any other criteria)...

   (a) Load balancing on ingestion

   ⇒ Assumptions made:

   (b) Load balancing on querying

   ⇒ Assumptions made:

   (c) I/O cost (i.e., minimum blocks flushed) on ingestion

   ⇒ Assumptions made:

   (d) I/O cost (i.e., minimum blocks retrieved) on querying

   ⇒ Assumptions made:

9. Given an empty **linear hash** with $f(x) = x$ (i.e., we directly apply the module to the keys), and a capacity of four keys per bucket, draw the result of inserting the following keys in the given order: 12, 4, 10, 49, 42, 60, 63, 53, 47, 27, 26, 28, 13, 52.

Theoretical questions

(a) Which is the main difference between the hash functions used in the linear hash and consistent hash algorithms?

(b) With respect to distributed systems, explain what is a distributed hash table (DHT), and provide a brief description of how consistent hashing guarantees balancing keys when adding new servers.