Amazon DynamoDB

Viet-Trung Tran

Amazon DynamoDB

- Simple interface
  - Key/value store
- Sacrifice strong consistency for availability
- “always writeable” data store
  - no updates are rejected due to failures or concurrent writes
- Conflict resolution is executed during read instead of write
- An infrastructure within a single administrative domain where all nodes are assumed to be trusted.

amazon
DynamoDB

Design consideration

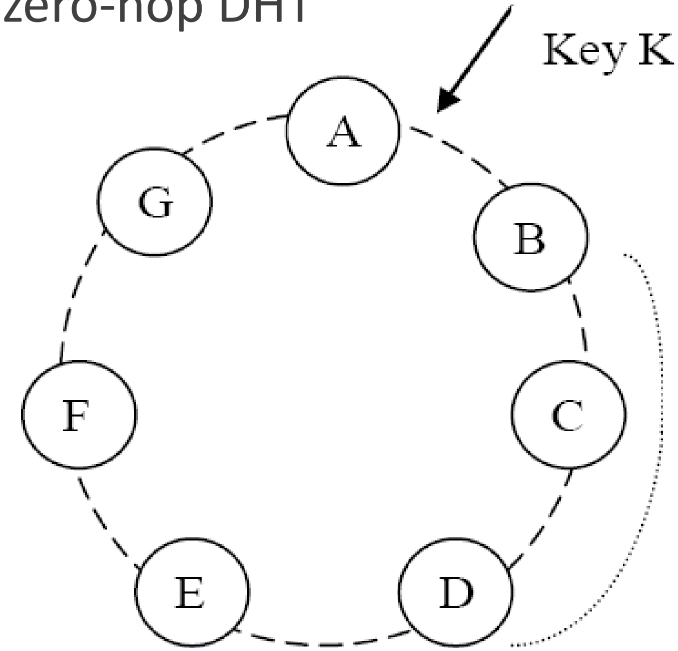
- Incremental scalability
- Symmetry
  - Every node in Dynamo should have the same set of responsibilities as its peers.
- Decentralization
  - In the past, centralized control has resulted in outages and the goal is to avoid it as much as possible
- Heterogeneity
  - This is essential in adding new nodes with higher capacity without having to upgrade all hosts at once

System architecture

- Partitioning
- High Availability for writes
- Handling temporary failures
- Recovering from permanent failures
- Membership and failure detection

Partition algorithm

- Consistent hashing: the output range of a hash function is treated as a fixed circular space or “ring”
- DynamoDB is a zero-hop DHT



Nodes B, C and D store keys in range (A,B) including K.

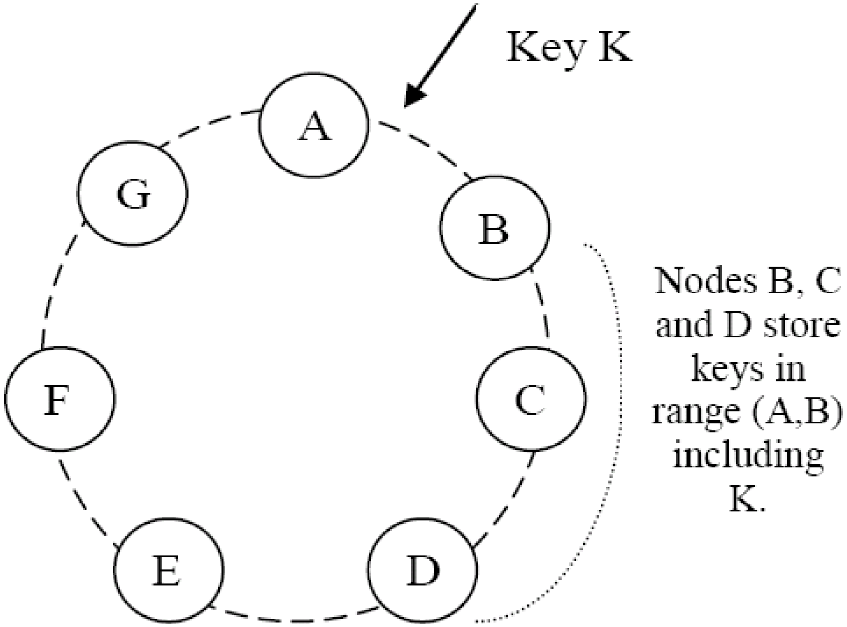
Grand challenge: every nodes must maintain an up-to-date view of the ring! How?

Virtual nodes

- Each node can be responsible for more than one virtual node.
  - Each physical node has multiple virtual nodes
  - More powerful machines have more virtual nodes
  - Distribute virtual nodes across the ring
- Advantages of using virtual nodes
  - If a node becomes unavailable, the load handled by this node is evenly dispersed across the remaining available nodes.
  - When a node becomes available again, or a new node is added to the system, the newly available node accepts a roughly equivalent amount of load from each of the other available nodes.
  - The number of virtual nodes that a node is responsible can decided based on its capacity, accounting for heterogeneity in the physical infrastructure.

Replication

- Each data item is replicated at N hosts.
  - N is the “preference list”: The list of nodes that is responsible for storing a particular key.

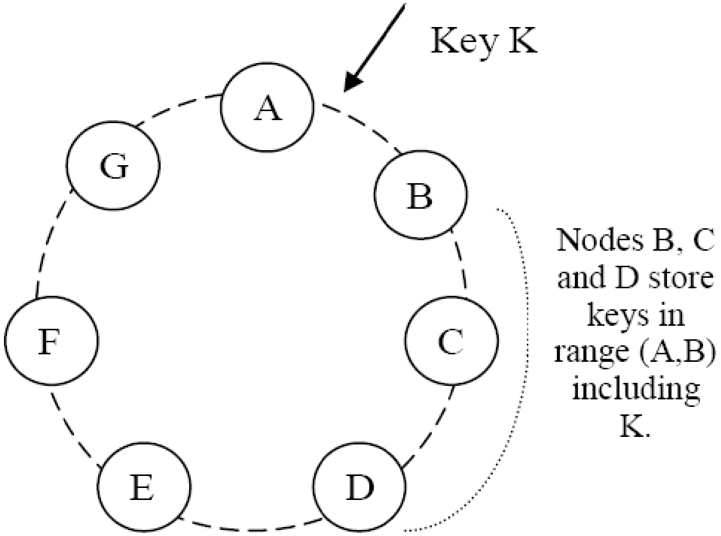


Quorum

- N: total number of replicas per each key/value pair
- R: minimum number of nodes that must participate in a sucessful reading
- W: minimum number of nodes that must participate in a sucessful writing
- Quorum-like system
  - R + W > N
  - In this model, the latency of a get (or put) operation is dictated by the slowest of the R (or W) replicas. For this reason, R and W are usually configured to be less than N, to provide better latency.

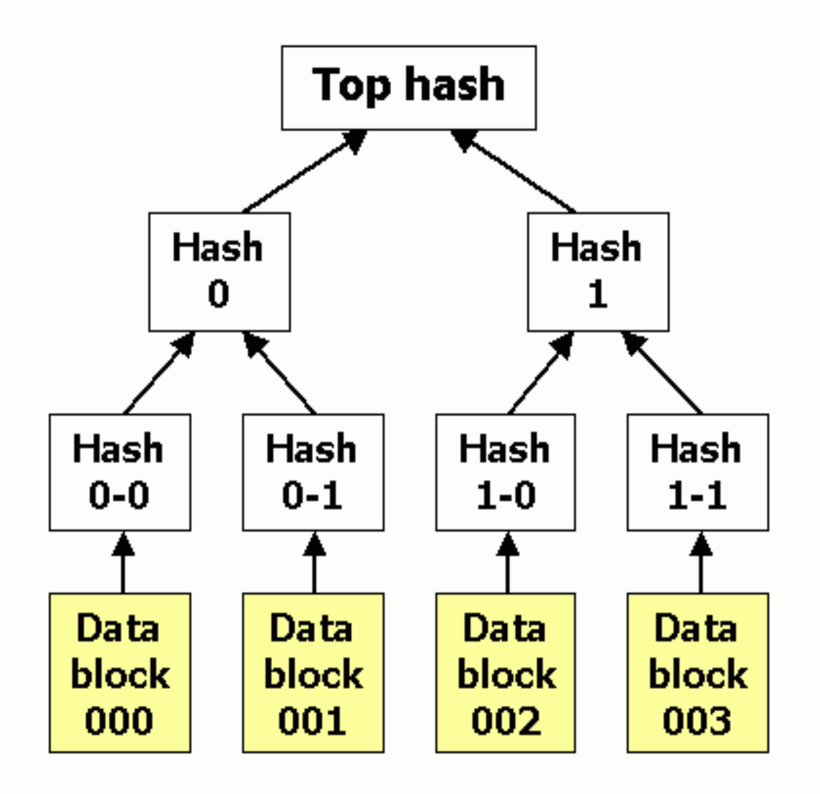
Temporary failures: Sloppy quorum and hinted handoff

- Assume N = 3. When B is temporarily down or unreachable during a write, send replica to E.
- E is hinted that the replica belongs to B and it will deliver to B when B is recovered.
- Again: “always writeable”



Replica synchronization

- Merkle tree
  - a hash tree where leaves are hashes of the values of individual keys
  - Parent nodes higher in the tree are hashes of their respective children
- Advantage of Merkle tree
  - Each branch of the tree can be checked independently without requiring nodes to download the entire tree
  - Help in reducing the amount of data that needs to be transferred while checking for inconsistencies among replicas



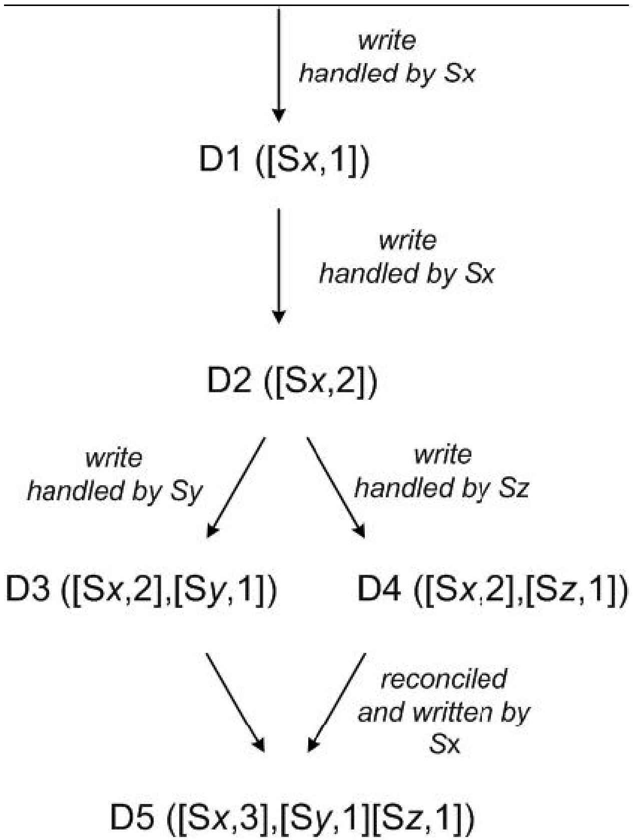
Data versioning

- A put() call may return to its caller before the update has been applied at all the replicas
- A get() call may return many versions of the same object.
- Key Challenge: distinct version sub-histories - need to be reconciled.
  - Solution: uses vector clocks in order to capture causality between different versions of the same object.

Vector clock

- A vector clock is a list of (node, counter) pairs.
- Every version of every object is associated with one vector clock.
- If the counters on the first object’s clock are less-than-or-equal to all of the nodes in the second clock, then the first is an ancestor of the second and can be forgotten.

Vector clock example



When the number of (node, counter) pairs in the vector clock reaches a threshold (say 10), the oldest pair is removed from the clock.

Technical summary

Problem	Technique	Advantage
Partitioning	Consistent Hashing	Incremental Scalability
High Availability for writes	Vector clocks with reconciliation during reads	Version size is decoupled from update rates.
Handling temporary failures	Sloppy Quorum and hinted handoff	Provides high availability and durability guarantee when some of the replicas are not available.
Recovering from permanent failures	Anti-entropy using Merkle trees	Synchronizes divergent replicas in the background.
Membership and failure detection	Gossip-based membership protocol and failure detection.	Preserves symmetry and avoids having a centralized registry for storing membership and node liveness information.

DynamoDB sum up

- Dynamo is a highly available and scalable data store for Amazon.com's e-commerce platform.
- Dynamo has been successful in handling server failures, data center failures and network partitions.
- Dynamo is incrementally scalable and allows service owners to scale up and down based on their current request load.
- Dynamo allows service owners to customize their storage system by allowing them to tune the parameters N, R,and W.

Thank you for your attention!

Q&A