Dynamo: Amazon's Highly Available Key-Value Store

Christopher Meiklejohn
Jiangnan Shangguan

Problem

- Reliability
  - shopping carts
  - session data
- Scalability
- Availability versus Consistency
  - during failure conditions

Requirements

- Query Model
  - key/value based
- Efficiency
  - 99.9th percentile
- Conflict Resolution
  - Always writable
  - Performed by application

System Architecture

- Partitioning
- Replication
- Versioning
- Membership
- Failure Handling
- Scaling

System Interface

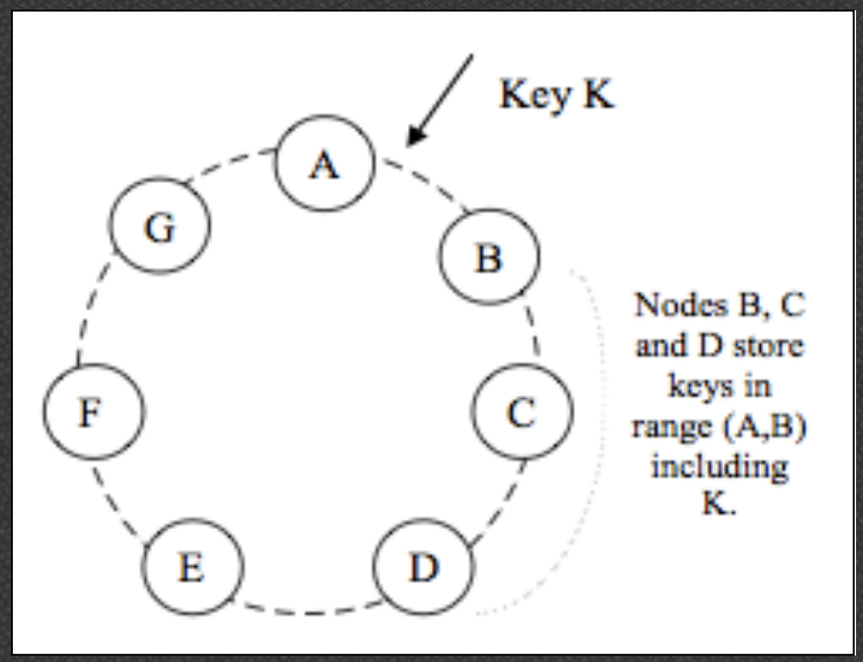
- get(key)
- put(key, context, object)
- MD5

Consistent Hashing

- Scale incrementally
  - Dynamic partitioning
- Ring
- Uniform data and load distribution
  - Virtual nodes

Replication

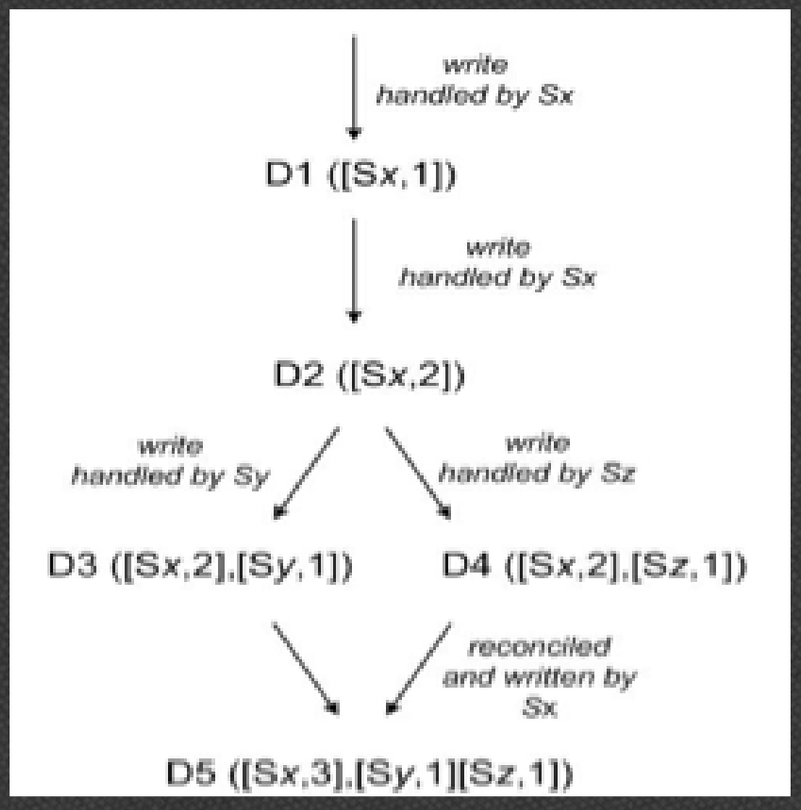
- Replicate at N virtual nodes
- Preference list (consisting of N virtual nodes)



G. DeCandia et al.,(2007) Dynamo: Amazon's highly available key-value store, in SOSP

Data Versioning

- Vector Clocks
  - Determine causality



G. DeCandia et al.,(2007) Dynamo: Amazon's highly available key-value store, in SOSP

Execution of GET/PUT

- Any node can be a coordinator
  - Coordinator responsible for writing v-clock
- Quorums (R/W)
  - R + W > N

Handling Failures

- Hinted Handoff
  - Metadata hint for owning node
- “Sloppy Quorums”
  - N healthy nodes
  - starting with preferences list

Replica Repair

- Hinted replicas no longer available
- Anti-entropy synchronization method
  - Merkle trees
    - Starting with partition key space
    - Nodes exchange trees corresponding to common partitions
- Implementation vague

Ring Membership

- Separate explicit mechanism
  - Nodes may be down for maintenance
  - Nodes may be powered up by accident
- Gossip protocol is used to propagate
- Nodes randomly select tokens mapping to virtual nodes for ownership.
  - Ownership transfer happens between nodes no longer responsible for that data

Implementation

- Three main components
  - Request coordination
  - Membership and failure detection
  - Local persistence

Local Persistence

- Pluggable backends
  - Memory backend
  - Berkeley DB
  - MySQL

Request Coordination

- Messaging substrate
- Similar to SEDA (pipelining)
- Heavy use of finite state machines
  - Read coordination
    - eg. send requests, wait for responses, fail, systematic resolution, read repair
  - Write coordination
- Latency requirements allow for any node to coordinate requests.

Resolution Strategies

- Business specific resolution
  - shopping cart
- Timestamp based resolution (ie. LWW)
  - session data

Performance

- Designed for commodity hardware
- Optimize performance using writer thread
  - Object cache
- Durable write quorum (DW)

Partitioning Strategies

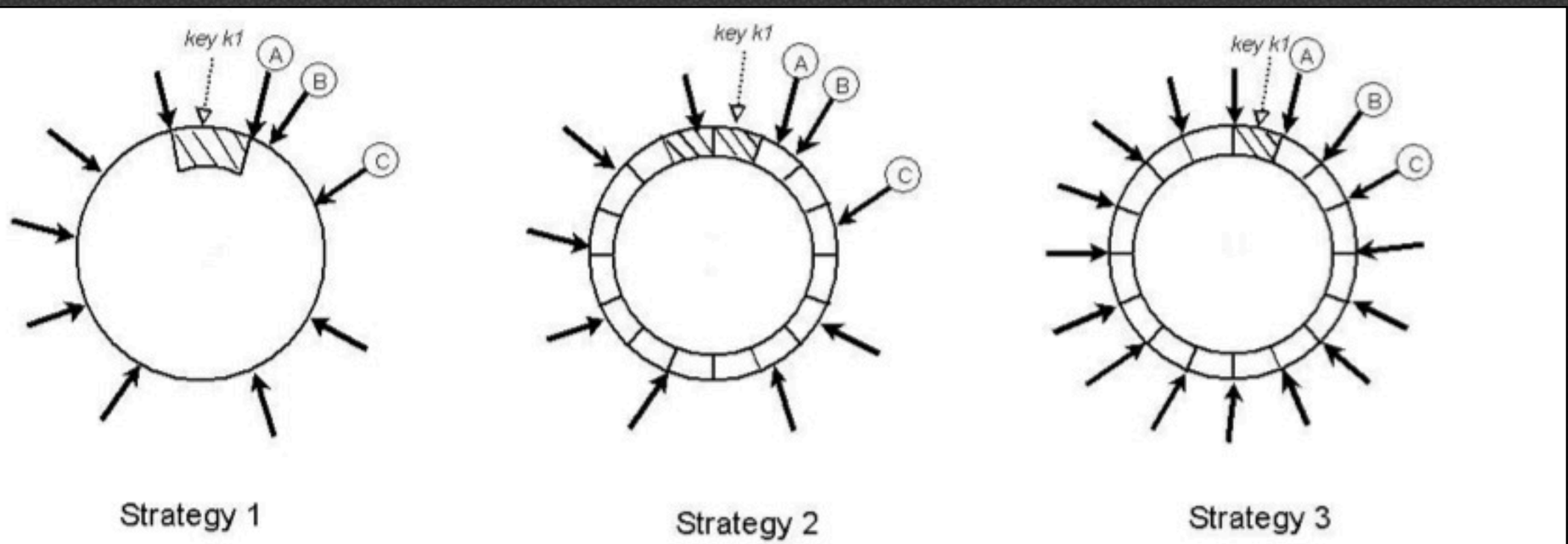


Figure 7: Partitioning and placement of keys in the three strategies. A, B, and C depict the three unique nodes that form the preference list for the key k1 on the consistent hashing ring (N=3). The shaded area indicates the key range for which nodes A, B, and C form the preference list. Dark arrows indicate the token locations for various nodes.

G. DeCandia et al.,(2007) Dynamo: Amazon's highly available key-value store, in SOSP

Divergence

- Failure scenarios
- High concurrency writes

Thank you!

Questions?