Dynamo

Amazon's Highly Available Key-value Store

Presented by: Cyber Infrastructure for Researchers Team

Tian Chen, Donovan Jones, Komal Kango, Jing Song & Kristi Perreault

Overview

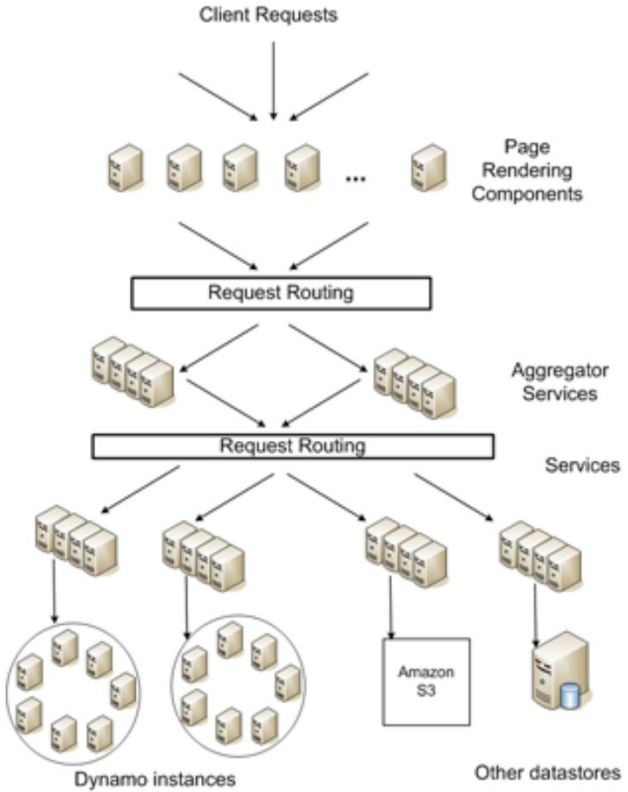
- **What is Dynamo?**
- Data Distribution
- Replication
- Solution for Inconsistencies
- Failure Handling
- Membership and Failure detection
- Experiences
- Summary

What is Dynamo?

Dynamo is a **highly available** primary-key only storage system which provides an “always on” experience.

Dynamo is the underlying **storage technology** for Amazon’s e-commerce platform (across multiple data centers) since it is able to **scale to peak load efficiency** without any downtime.

Amazon's Requirements for Dynamo



99.9 percentile tail latency

Figure 1 from Dynamo Paper

System Simplifications

- Query Model
  - Simple Read and Write operations to data item that is identified by a unique key
- Weaker consistency to achieve higher availability

Key Design Aspects

- Eventually consistent data store
  - All updates reach all replicas eventually
- Highly available for Writes
- Conflict Resolution is done by Application

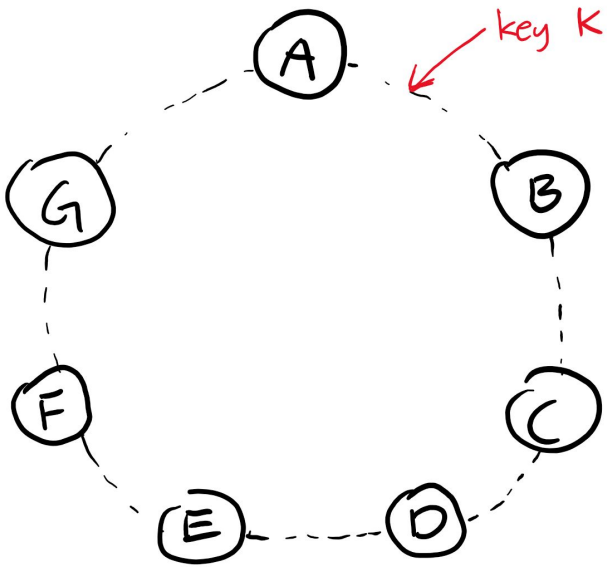
Other Design Aspects

- Incremental scalability
- Symmetry
- Decentralization
- Heterogeneity

Overview

- What is Dynamo?
- **Data Distribution**
- Replication
- Solution for Inconsistencies
- Failure Handling
- Membership and Failure detection
- Experiences
- Summary

Partition Algorithm: Consistent Hashing



Why Consistent Hashing?

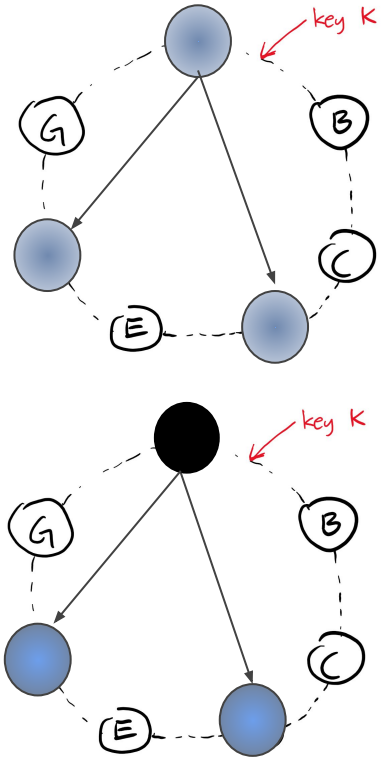
- Remap Less Keys

Problems with Typical Consistent Hashing

- Randomly assign node to a position→ non-uniform distribution
- Does not account for performance heterogeneity

Dynamo’s Solution for Consistent Hashing

Handling unable nodes:



Virtual Nodes

- Map one node to multiple positions in the ring

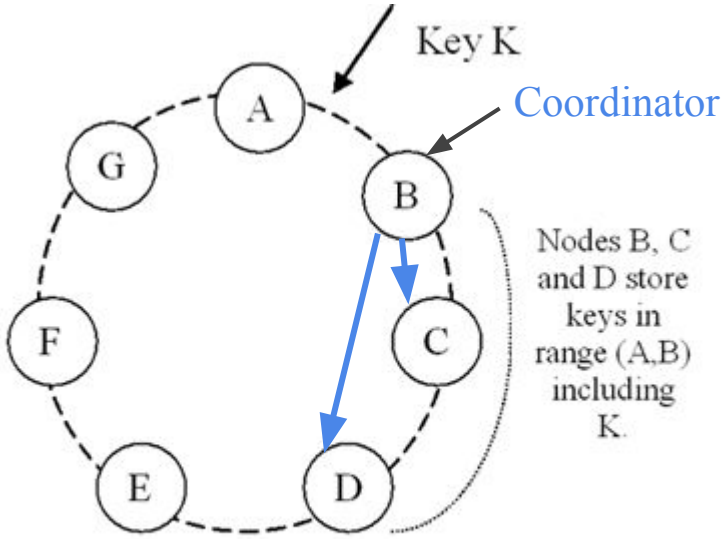
Advantages

- Load distributed uniformly on the available node
- New node receives equivalent load from the available nodes
- Heterogeneity

Overview

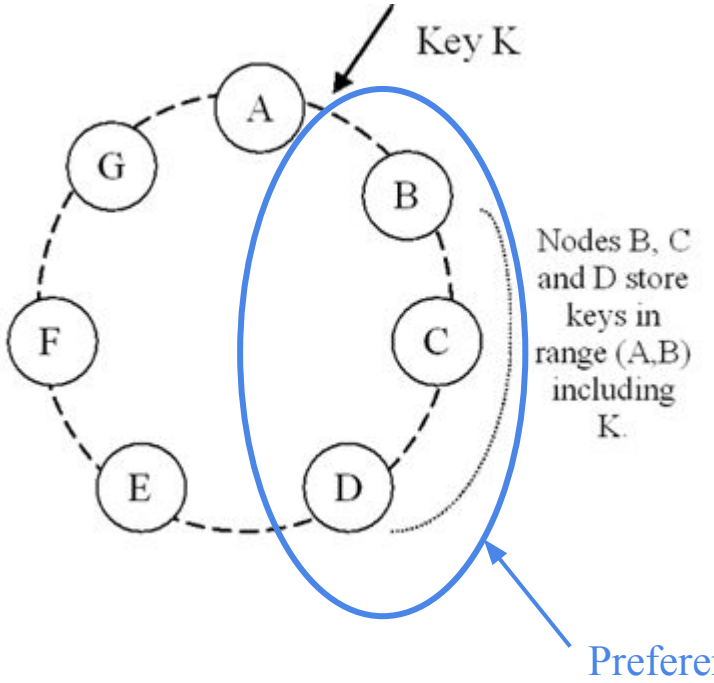
- What is Dynamo?
- Data Distribution
- **Replication**
- Solution for Inconsistencies
- Failure Handling
- Membership and Failure detection
- Experiences
- Summary

Replication



- N: a system parameter indicating how many replicas should be made
- Suppose N = 3

Replication



- Preference List: the list of nodes that are responsible for storing a particular key

Overview

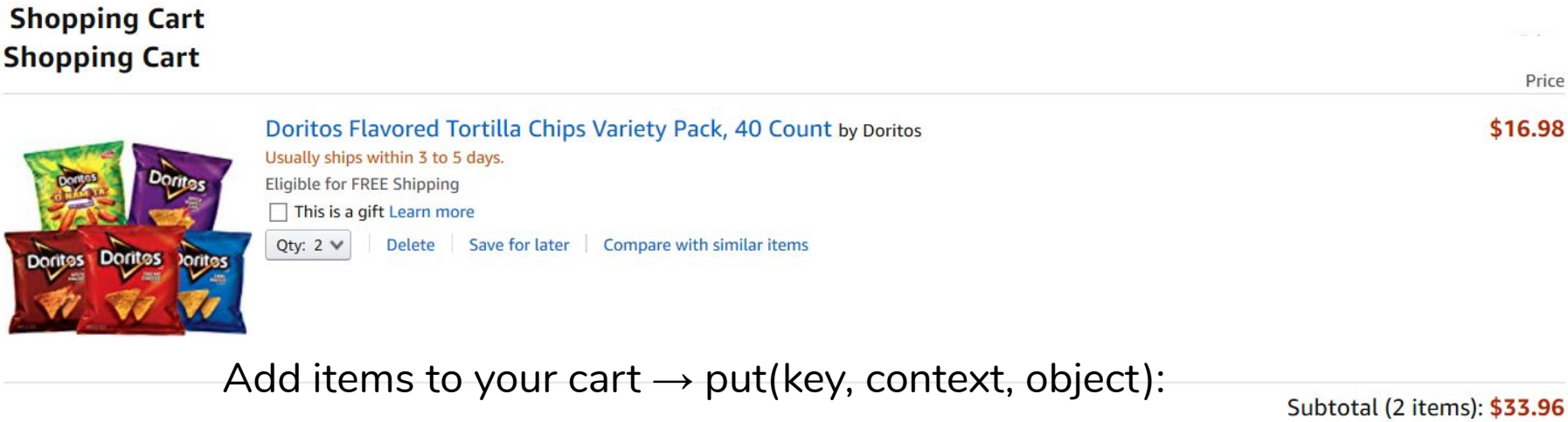
- What is Dynamo?
- Data Distribution
- Replication
- **Solution for Inconsistencies**
- Failure Handling
- Membership and Failure detection
- Experiences
- Summary

This should be familiar...

Check out your cart → get(key):

- Finds the object replicas based on the key
- Returns the object or a list of objects with conflicting versions, together with a context
- Context: encoded system metadata about the object

Craving for Doritos…



Add items to your cart → put(key, context, object):

- Determines where to put the replicas of the object
- Writes the replicas to disk

Syntactic Reconciliation

- Later version subsumes the earlier one
- System could determine the authoritative version automatically

Sister (in California) ’s Suggestion…

Shopping Cart

Price

Coca-Cola Soda Soft Drink, 8.5 fl oz, 12 Pack by Coca-Cola — $21.99

In stock on April 10, 2020.

Eligible for FREE Shipping

☐ This is a gift Learn more

Qty: 1 | Delete | Save for later | Compare with similar items

Doritos Flavored Tortilla Chips Variety Pack, 40 Count by Doritos — $16.98

Only 1 left in stock (more on the way).

Eligible for FREE Shipping

☐ This is a gift Learn more

Qty: 2 | Delete | Save for later | Compare with similar items

Save 5% now with Subscribe & Save ›

Subtotal (3 items): $55.95

Brother (in Arizona) ’s suggestion…

Shopping Cart

Price

Diet Pepsi Soda, 7.5 Ounce Mini Cans, 10 Pack by Pepsi

In stock on April 21, 2020.

Eligible for FREE Shipping

☐ This is a gift Learn more

Qty: 1 | Delete | Save for later | Compare with similar items

$3.99

Save 5% now with Subscribe & Save ›

Doritos Flavored Tortilla Chips Variety Pack, 40 Count by Doritos

Only 1 left in stock (more on the way).

Eligible for FREE Shipping

☐ This is a gift Learn more

Qty: 2 | Delete | Save for later | Compare with similar items

$16.98

Save 5% now with Subscribe & Save ›

Subtotal (3 items): $37.95

Semantic Reconciliation

- Conflicting Versions
- Client applications have to manually perform the reconciliation
  - Amazon: merge all versions for largest profit

Data Versioning: Vector Clocks

- How the system handles multiple, conflicting branches of data evolution
- Vector Clocks: a list of (node, counter) pairs that is associated with every version of every object

Vector Clocks

Shopping Cart

Price

Shopping Cart

Price

Doritos Flavored Tortilla Chips Variety Pack, 40 Count by Doritos

$16.98

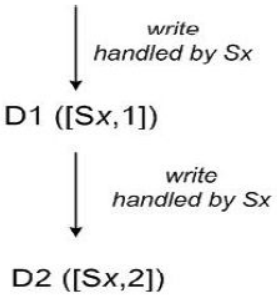
Usually ships within 3 to 5 days.

Eligible for FREE Shipping

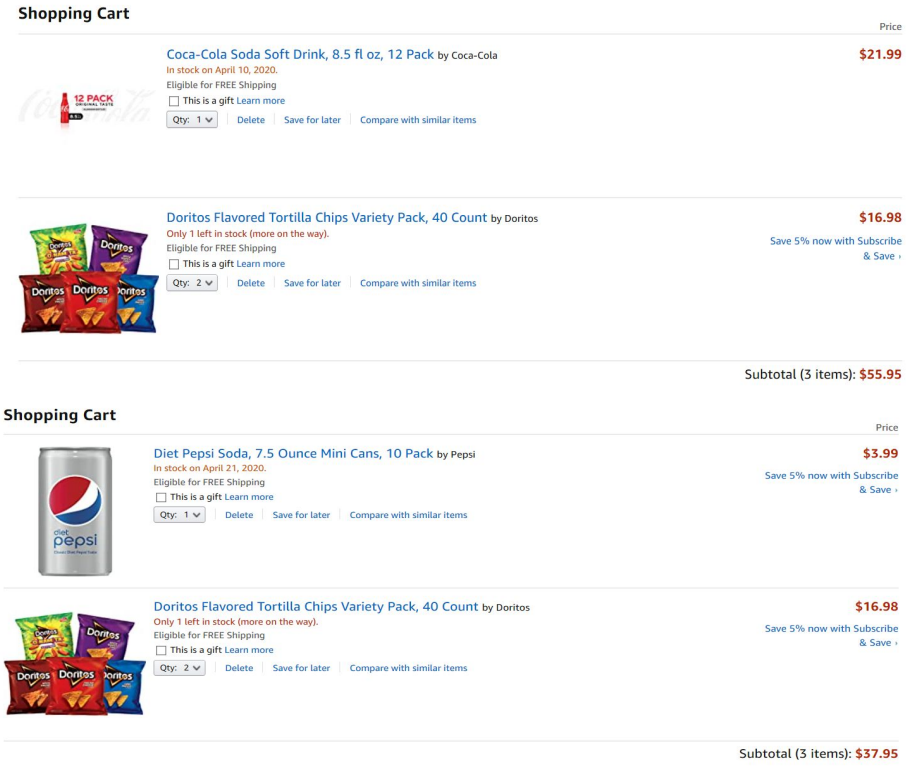
☐ This is a gift Learn more

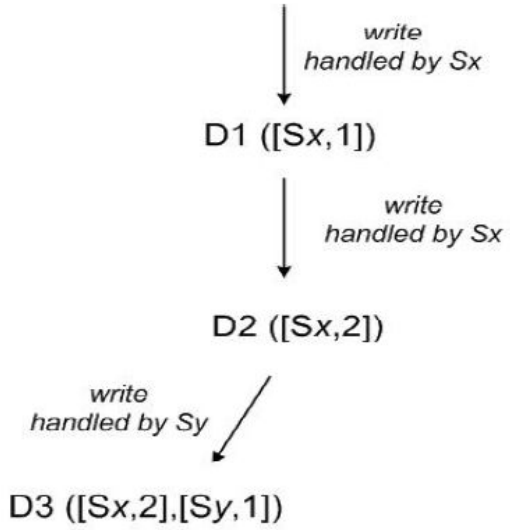
Qty: 2 | Delete | Save for later | Compare with similar items

Subtotal (2 items): $33.96

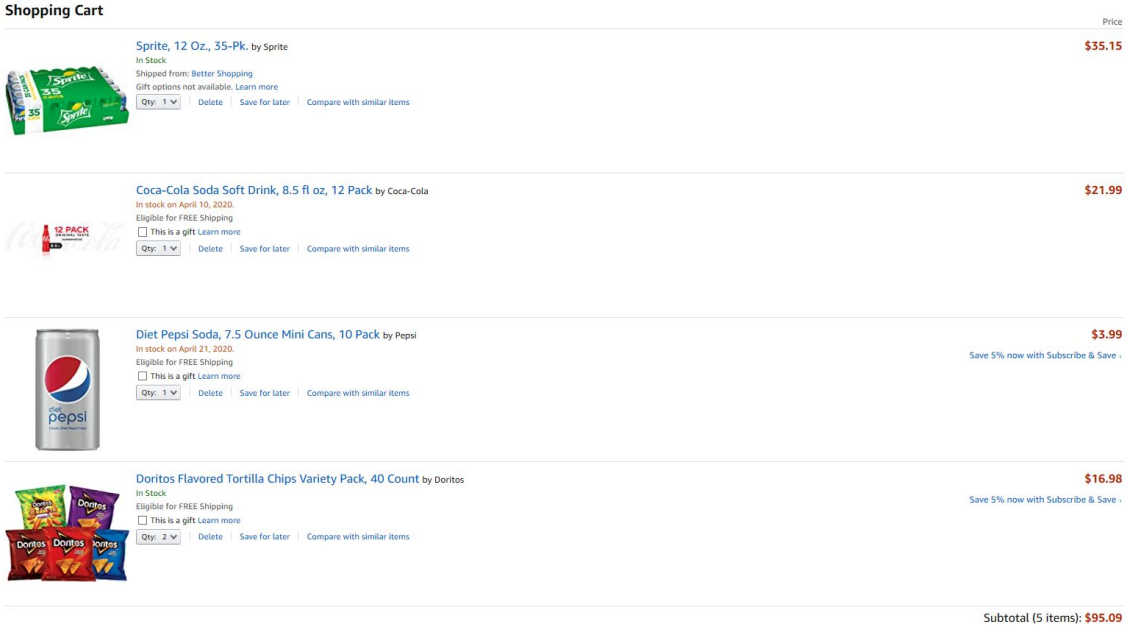


Vector Clocks





Vector Clocks



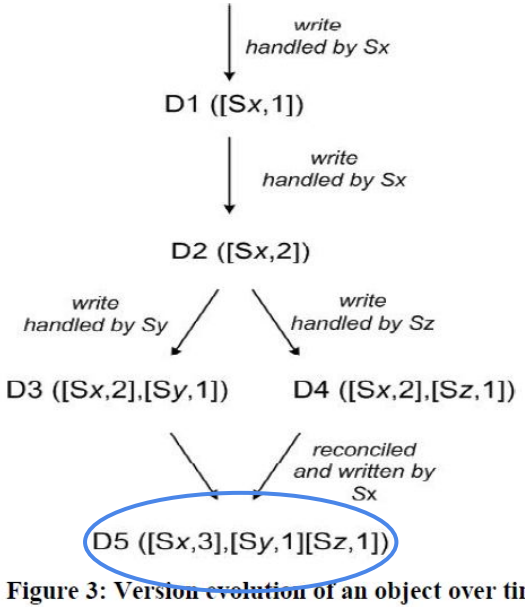


Figure 3: Version evolution of an object over time.

Problems with Vector Clocks

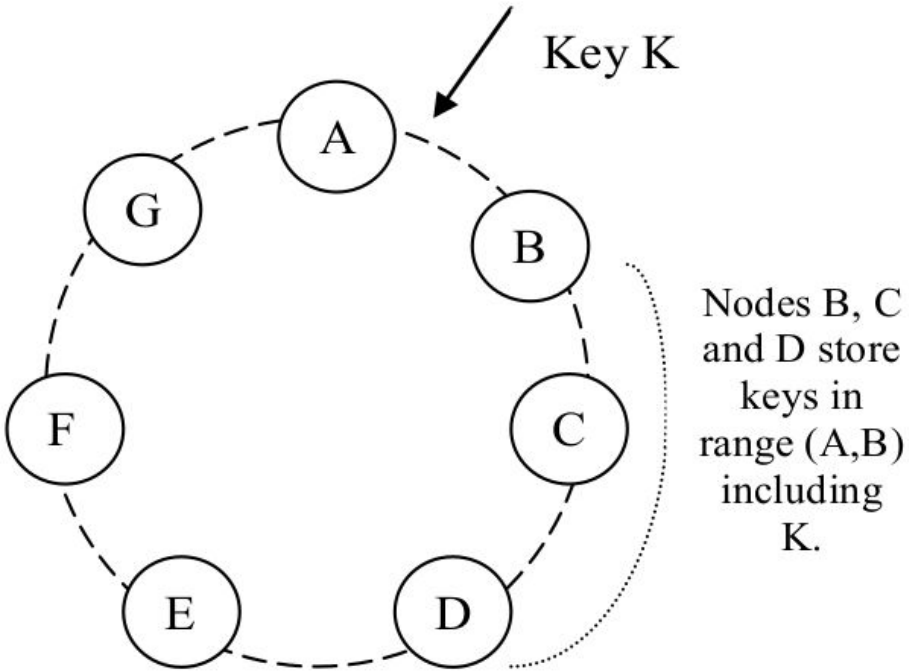
- In practice, write requests are not always handled by the top N nodes in the preference list
- The size of vector clocks will grow
- Dynamo’s solution: Clock Truncation Scheme

Sloppy Quorum: Introducing the problem

How to maintain the consistency among replicas during put() and get() operation?

How can we design the strategy so that we can retrive the same data from any replicas?

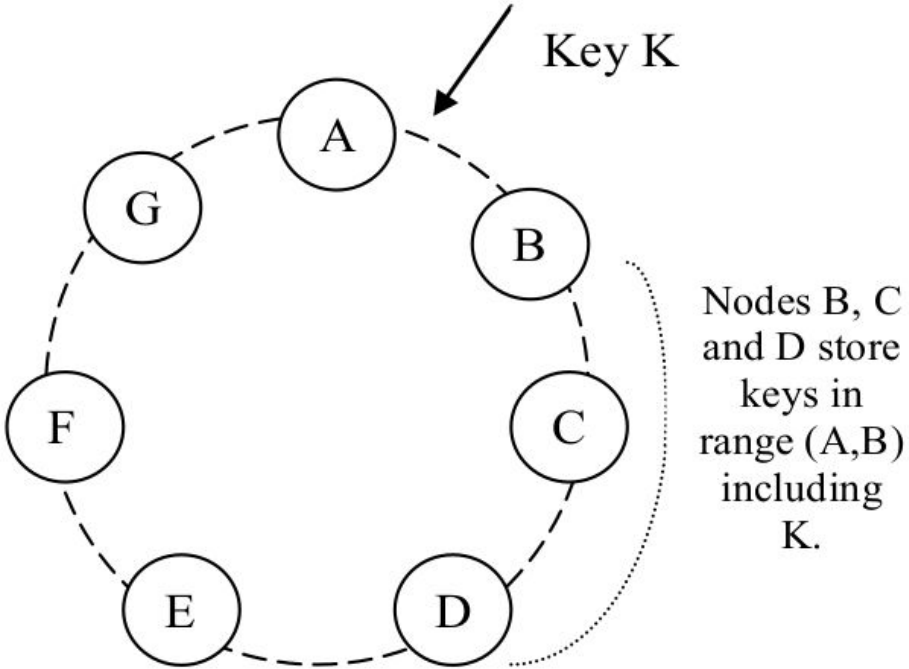
what is the most intuitive idea?



Read-One-Write-All:

Send the data to all replicas
Read from one of them

Congratulations! We have realized a consistency model:



Problems:

Write is vulnerable to be affected by node failure

Improvement:

A better model for replicas control

What is Quorum?

A quorum is the minimum number of votes that a distributed transaction has to obtain in order to be allowed to perform an operation in a distributed system.
A quorum-based technique is implemented to enforce consistent operation in a distributed system.

Quorum protocol:

This protocol has three values:

R, W and N.

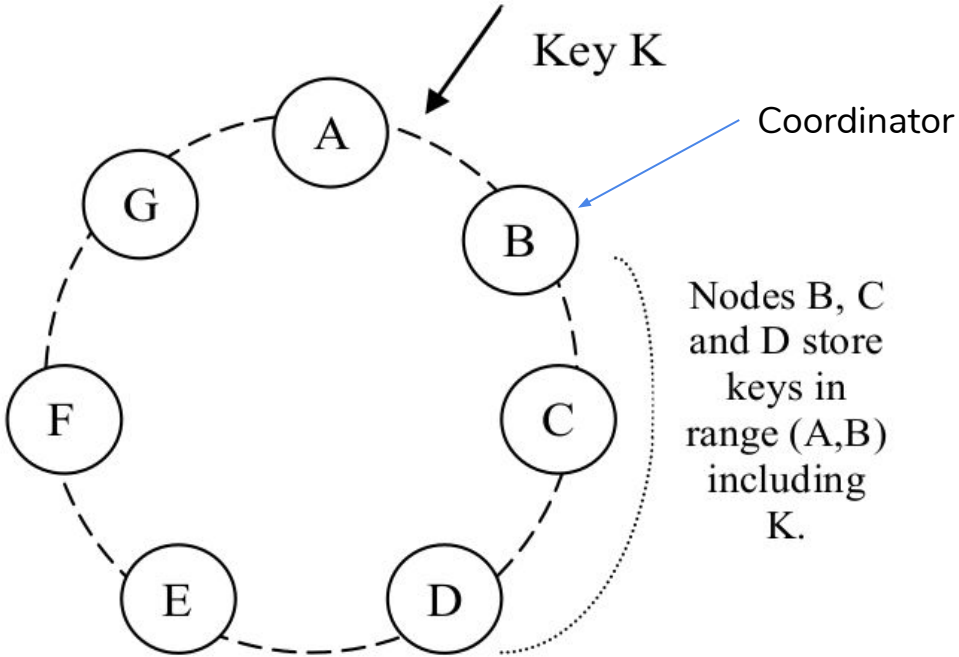
R is the minimum number of nodes that must participate in a successful read operation.
W is the minimum number of nodes that must participate in a successful write operation.
N is the number of nodes that being involved in read or write operation.

Quorum protocol:

Setting R and W such that R + W > N yields a quorum-like system.

In this model, the latency of a read (or write) operation is dictated by the slowest of the R (or W) replicas.

Quorum protocol in Dynamo in order to maintain the consistency



A request(read or write) involves the first N healthy nodes

Coordinator(the first among the top N healthy nodes) will process the requests and maintain the consistency among its replicas

put() operation:

For a put() request:

1. The coordinator generates the new version (using vector lock)
2. The coordinator writes the new version locally.
3. The coordinator sends the new version (along with the new vector clock) to the N highest-ranked reachable nodes.
4. The write is considered successful if at least W-1 nodes respond .

get() operation:

Similarly, for a get() request:

1. The coordinator requests all existing versions of data for that key from the N highest-ranked reachable nodes in the preference list for that key.
2. The coordinator waits for R responses before returning the result to the client.
3. If the coordinator ends up gathering multiple versions of the data, it returns all the versions it deems to be causally unrelated.
4. The divergent versions are then reconciled and the reconciled version superseding the current versions is written back.

Why it is called sloppy quorum?

Traditional quorum would be unavailable during server failures and network partitions.

Dynamo does not enforce strict quorum membership.

Instead, it uses a “sloppy quorum”: all read and write operations are performed on the first N healthy nodes from the preference list, which may not always be the first N nodes encountered while walking the consistent hashing ring.

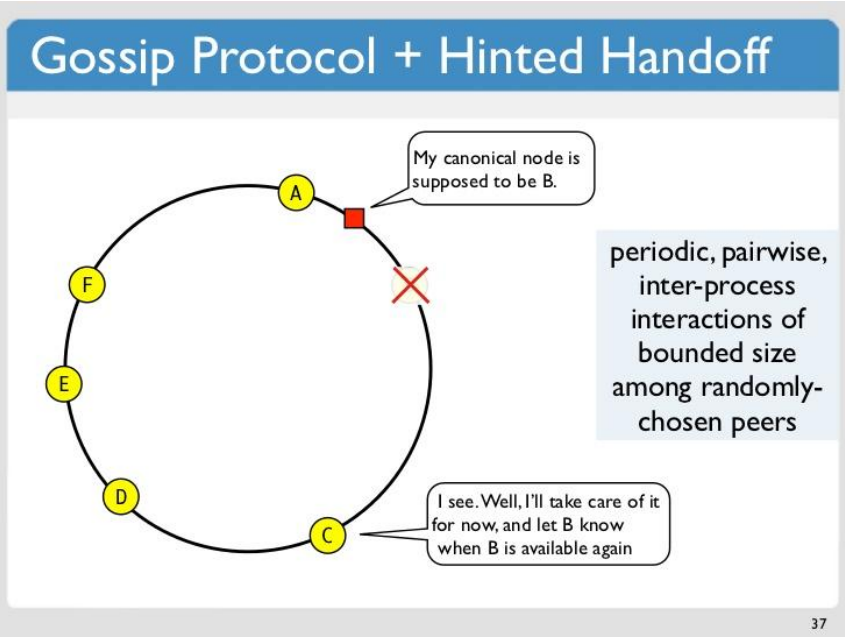
The weakness of Quorum:

- Dirty read : it should try to be avoided
- Eventual consistency: in some cases we need strong consistency

Overview

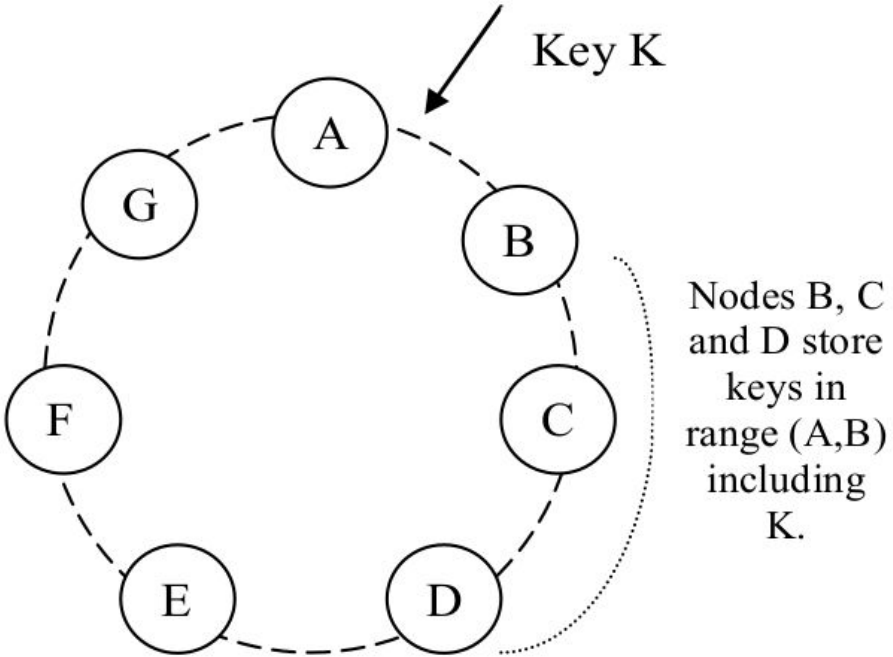
- What is Dynamo?
- Data Distribution
- Replication
- Solution for Inconsistencies
- **Failure Handling**
- Membership and Failure detection
- Experiences
- Summary

Failure handling: Hinted handoff

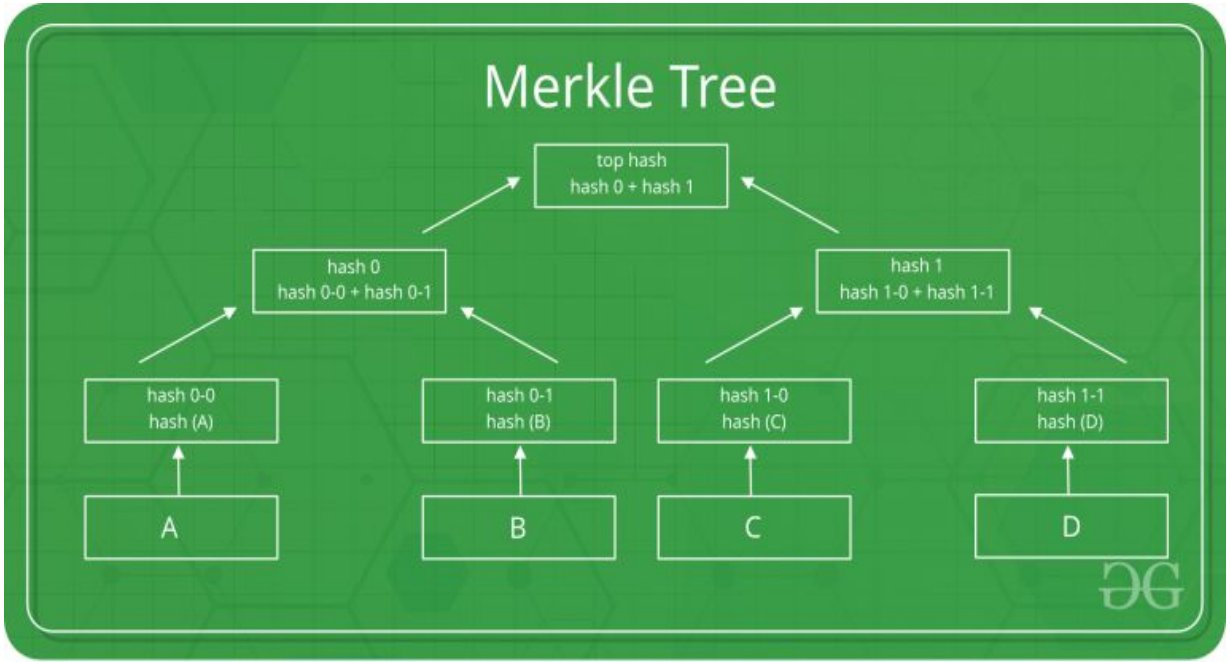


- Sloppy quorum
- Data stays with neighbor node if home node is down
- Data returns to home node once it is working again
- Ensures durability

Failure handling: Hinted handoff example



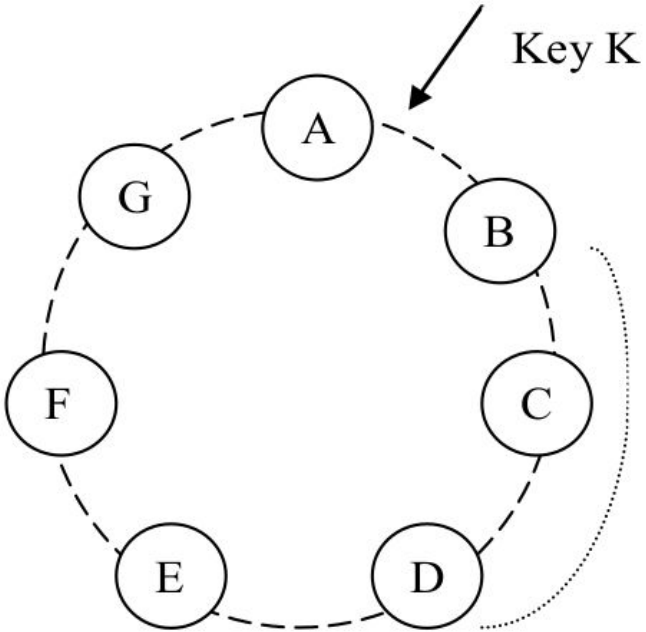
Handling permanent failures: Replica synchronization



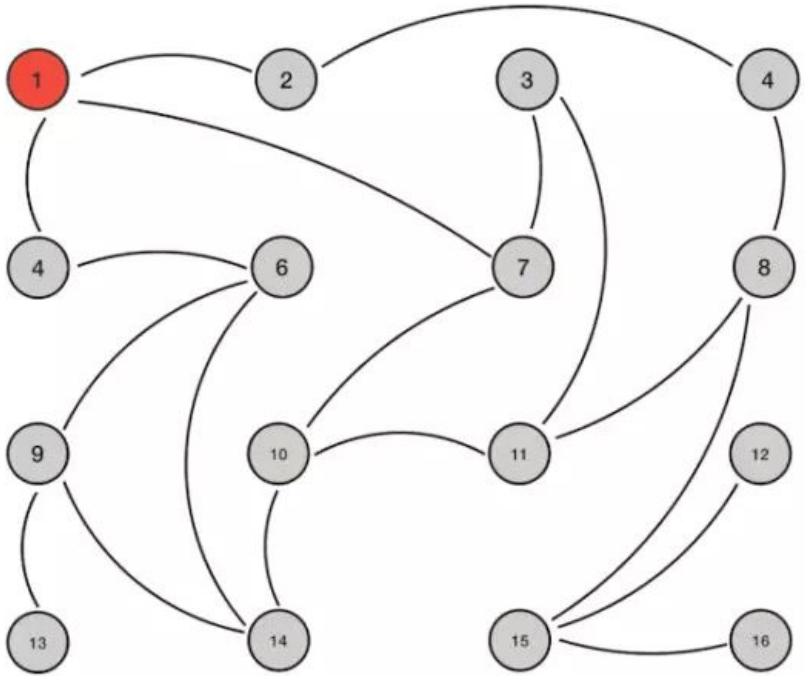
Overview

- What is Dynamo?
- Data Distribution
- Replication
- Solution for Inconsistencies
- Failure Handling
- **Membership and Failure detection**
- Experiences
- Summary

Membership and Failure Detection



Membership and Failure Detection



Membership and Failure Detection

- Problem with this method:
  - Logically partitioned ring

- Dynamo’s Solution:
  - Some nodes are seeds

Overview

- What is Dynamo?
- Data Distribution
- Replication
- Solution for Inconsistencies
- Failure Handling
- Membership and Failure detection
- **Experiences**
- Summary

Reconciliation

- Business Logic Specific Reconciliation
  - Amazon Shopping Cart

- Timestamp Based Reconciliation
  - Maintaining customer sessions

Performance & Durability

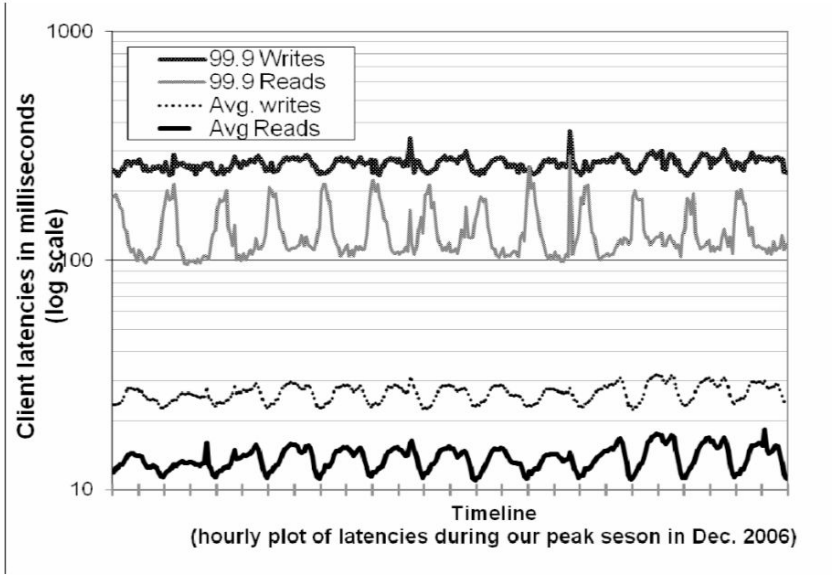


Figure 4

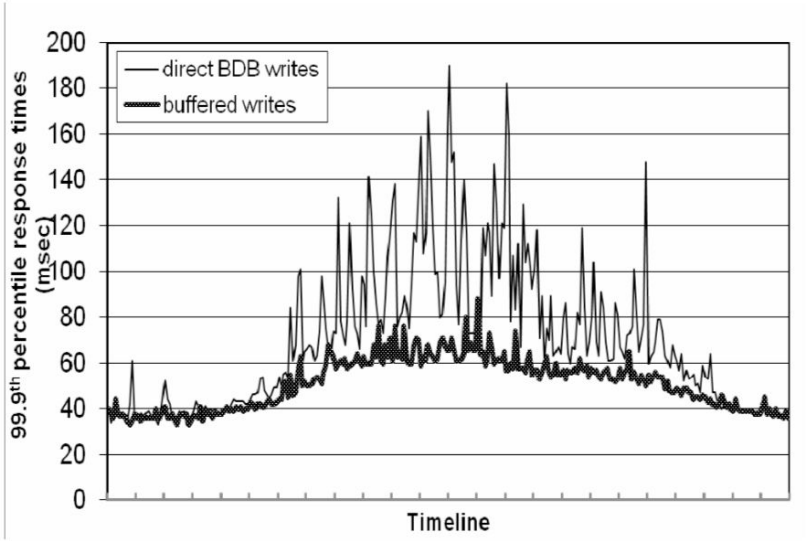


Figure 5

Divergent Versions

- Shopping cart service profiled for 24hrs
  - 99.4% of requests saw one version
  - 0.00057% of requests saw 2 versions
  - 0.00047% of requests saw 3 versions
  - 0.00009% of requests saw 4 versions

- Divergent versions created **rarely**

Client & Server Coordination

- Client-driven
  - Load balancer not required
  - Fair load distribution guaranteed
- Pull approach
  - Better scalability
  - Less maintenance
- Latency significantly less than server coordination

	99.9th percentile read latency (ms)	99.9th percentile write latency (ms)	Average read latency (ms)	Average write latency (ms)
Server-driven	68.9	68.5	3.9	4.02
Client-driven	30.4	30.4	1.55	1.9

Table 2

Overview

- What is Dynamo?
- Data Distribution
- Replication
- Solution for Inconsistencies
- Failure Handling
- Membership and Failure detection
- Experiences
- **Summary**

Summary

- Dynamo is a highly available and scalable data store
- Data partitioning with Consistent Hashing
- Vector clocks allow for high availability
- Temporary Failures
  - Sloppy Quorum
  - Hinted Handoff
- Permanent Failures
  - Anti-entropy and Merkle Trees
- Membership and failure detection
  - Gossip-based protocol

Where is Dynamo Now?

- Data store in Amazon Shopping Cart
- Amazon Web Services DynamoDB
  - NoSQL database service
  - BUT centralized
- Same principles as Dynamo

Questions?