Getting Started with DynamoDB

Ivan Mushketyk

@mushketyk brewing.codes

Overview

- Basic concepts of DynamoDB
- How to access stored data
- DynamoDB internals
- DynamoDB limitations
- What AWS charges for
- Using the AWS console

DynamoDB Introduction

Main DynamoDB concepts (tables, items, attributes)

DynamoDB types

Keys in DynamoDB

Ways to query data

Table, Items, Attributes

```
{
  “Id”: “1234”,
  “Name”: “Bruce”,
  “Title”: “Mr”
},
{
  “Id”: “2345”,
  “Name”: “Robin”
}
```

```
“Name”: “Bruce”
“Age”: 30
“Photo”: “f35D…g7AA”
“Protagonist”: true
“SuperPowers”: null
```

- String
- Age
- Binary
- Boolean
- Null

```
“List”: [1,”Batmobile”,true]

“Friends”: [“Robin”, “Alfred”, “Rachel”,
“Gordon”]

“Images”: [“3D..fe”, “45f…4”]

”Numbers”: [1, 2, 4, 10]

”Address”: {

  “Street”: “1007 Mountain Drive”,

  “City”: ”Gotham”

}
```

- List (any type)
- Set of Strings
- Set of Binaries
- Set of Numbers
- Nested objects

Key Types

Simple key

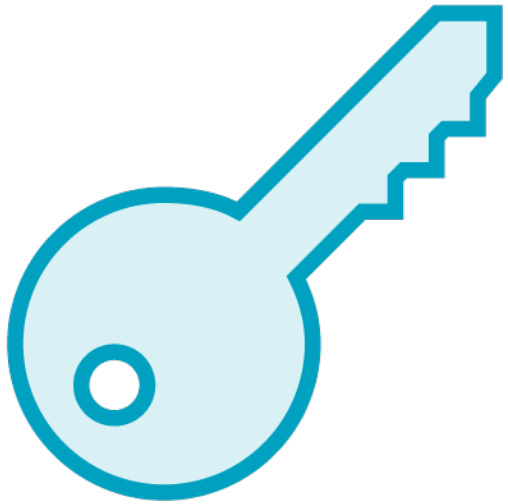
Partition key

Composite key

Partition and sort keys

Key attributes should be decided in advance and can’t be changed.

Forum Users



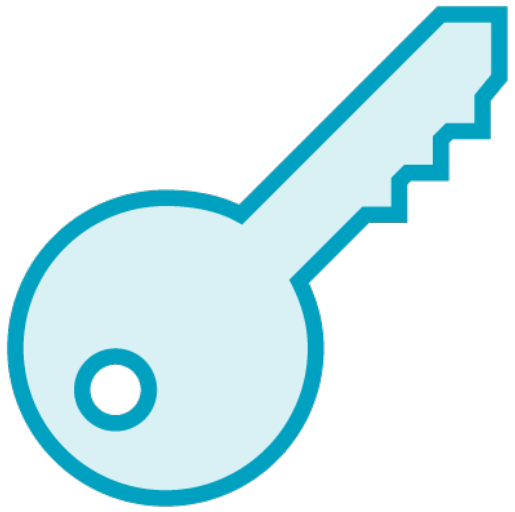
- Have a table of users
- Every user have unique ID
- Query to get a user by ID
- Partition key allows to get one item by ID

Data Example

Partition key

ID	Email	Name
1	joe@example.com	Joe
2	peter@example.com	Peter
3	anonymous@example.com	Kyle

Forum Posts



Have a table of posts

- UserID
- Timestamp

Need to find all posts by one user

Sort them by time

Can’t do with partition key

Query Example

Partition key

UserID	Timestamp (Sort Key)	Message
1	1498916052	Lunch time?
1	1498915072	Just had my sandwich
1	1498928631	This forum is boring....
2	1498163954	Hello everybody
3	1497009531	Kittens photos here:

Sort key

Pair of partition and sort key should be unique

How to Query Composite Key

Provide both partition and sort key values

Provide partition key value and:

- Use operators like <=, >=
- Use BETWEEN operator
- Use begins_with function
- Provide filter expression for other attributes

Can retrieve up to 1MB of data

Scans

Can provide arbitrary search expressions

High price

- Much slower
- More expensive

Use keys instead

Scans are last resort

Scaling DynamoDB

DynamoDB scaling model

RCU/WCU

How do you scale DynamoDB

Scaling in Regular DB

Determine how many servers do we need

Add more servers if necessary to process more requests

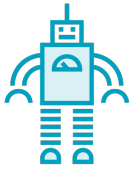
Indirect process

Scaling in DynamoDB

Directly specify how many requests you will have

Define number of RCUs and WCUs

Can change at any time



Supports auto-scaling

What if We Under-provision

Burst capacity

DynamoDB will return an error

Can retry a request if a spike in requests

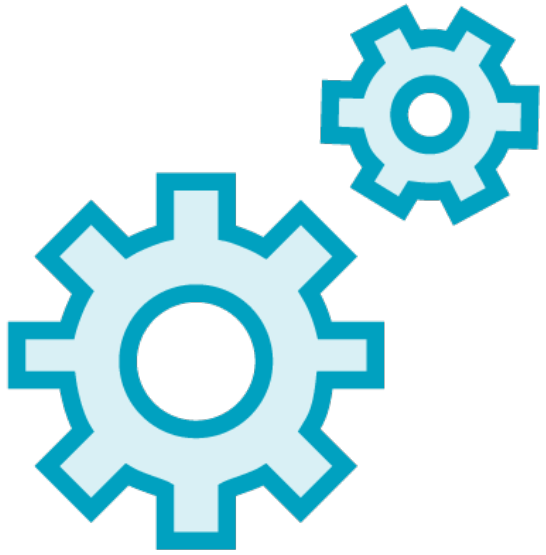
Demo

Go to AWS console

Create a table

Perform several queries

DynamoDB Internals



DynamoDB under the hood

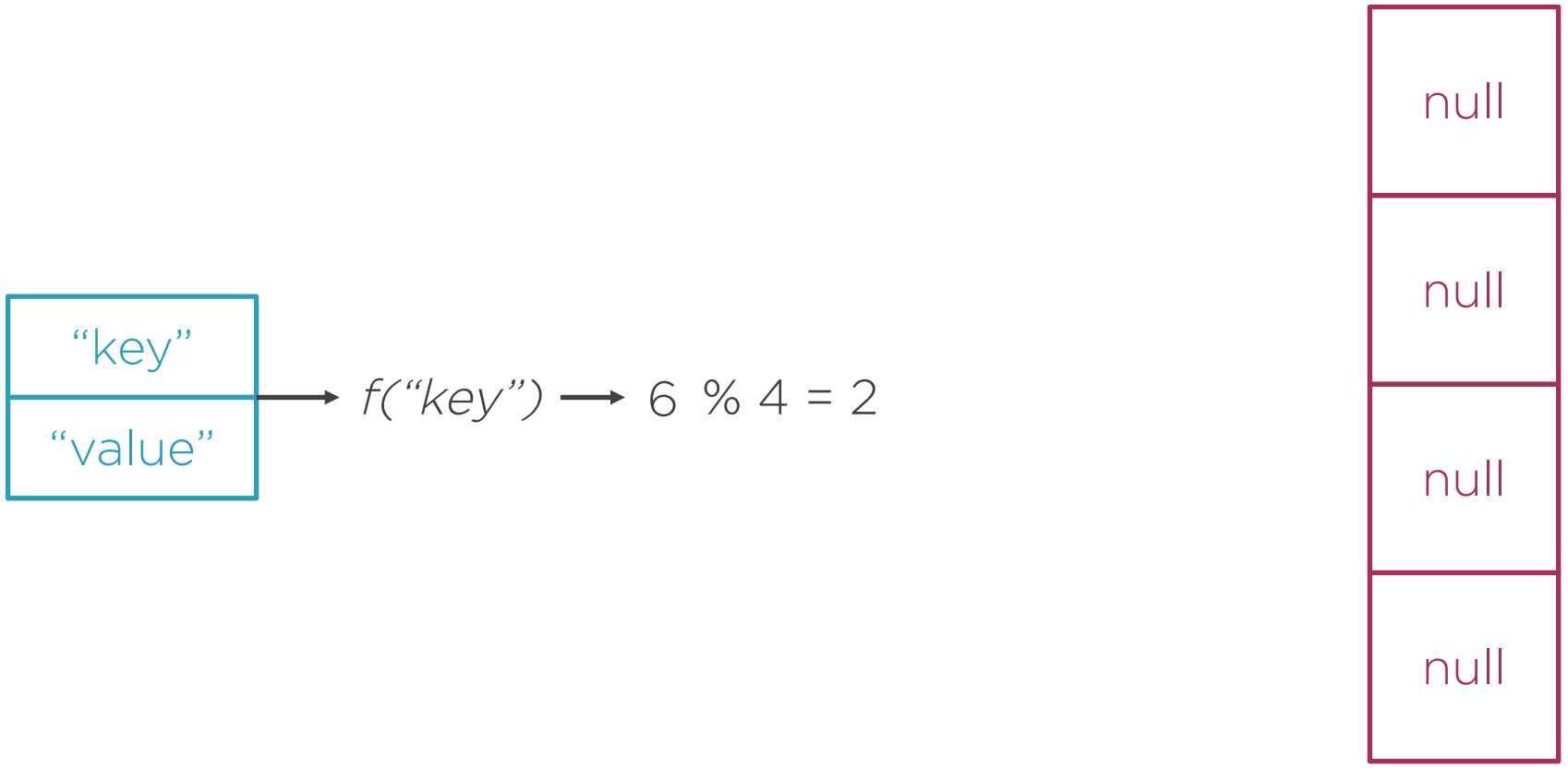
Consistent hashing

Partitions in DynamoDB

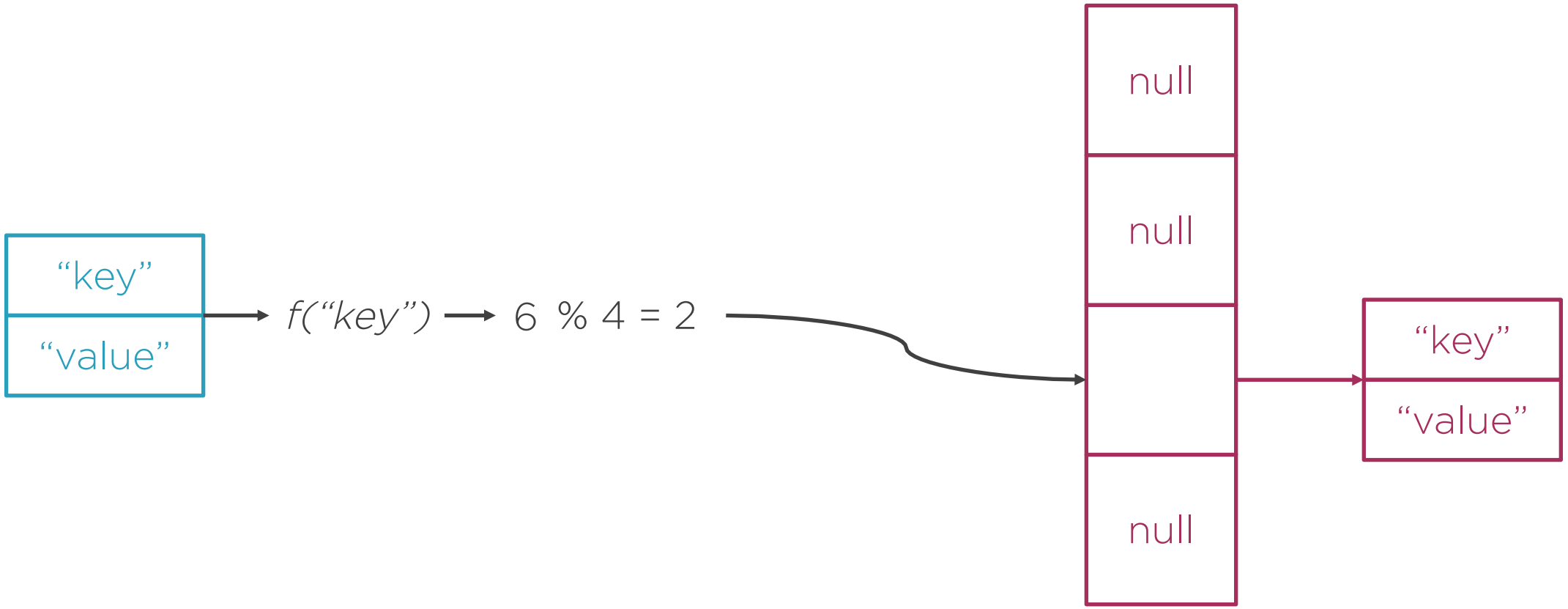
How keys relate to partitions

Understand limitations and features of DynamoDB

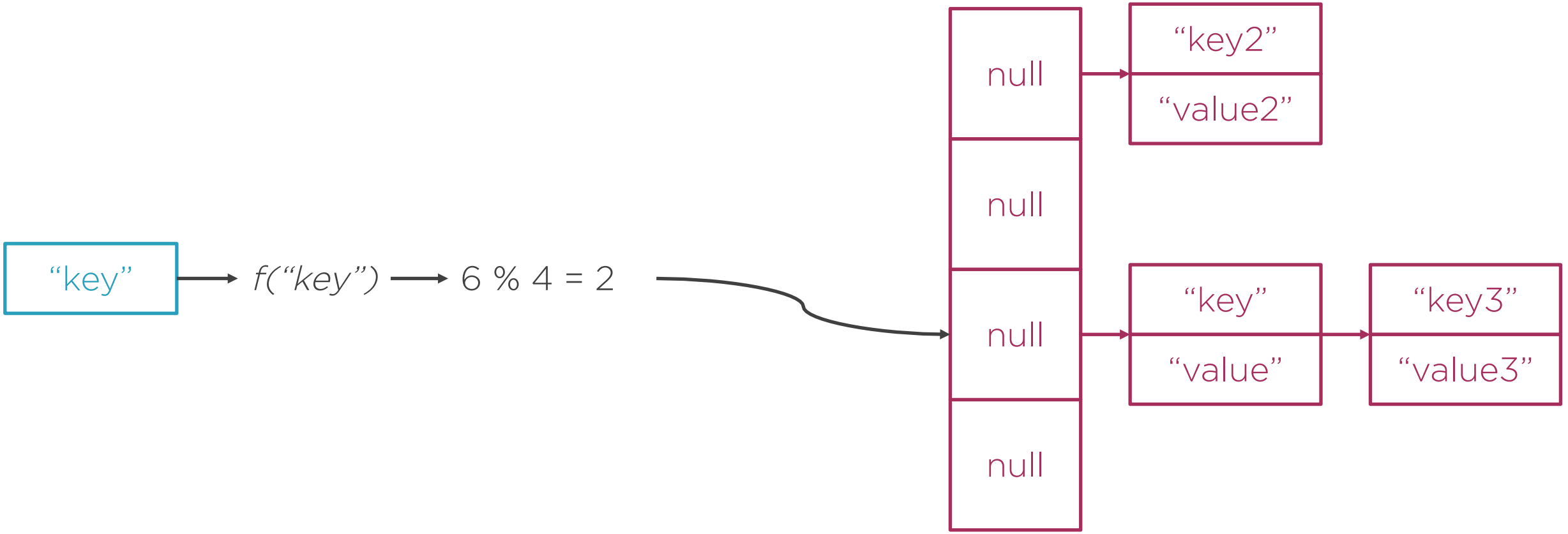
Hash Table Recap



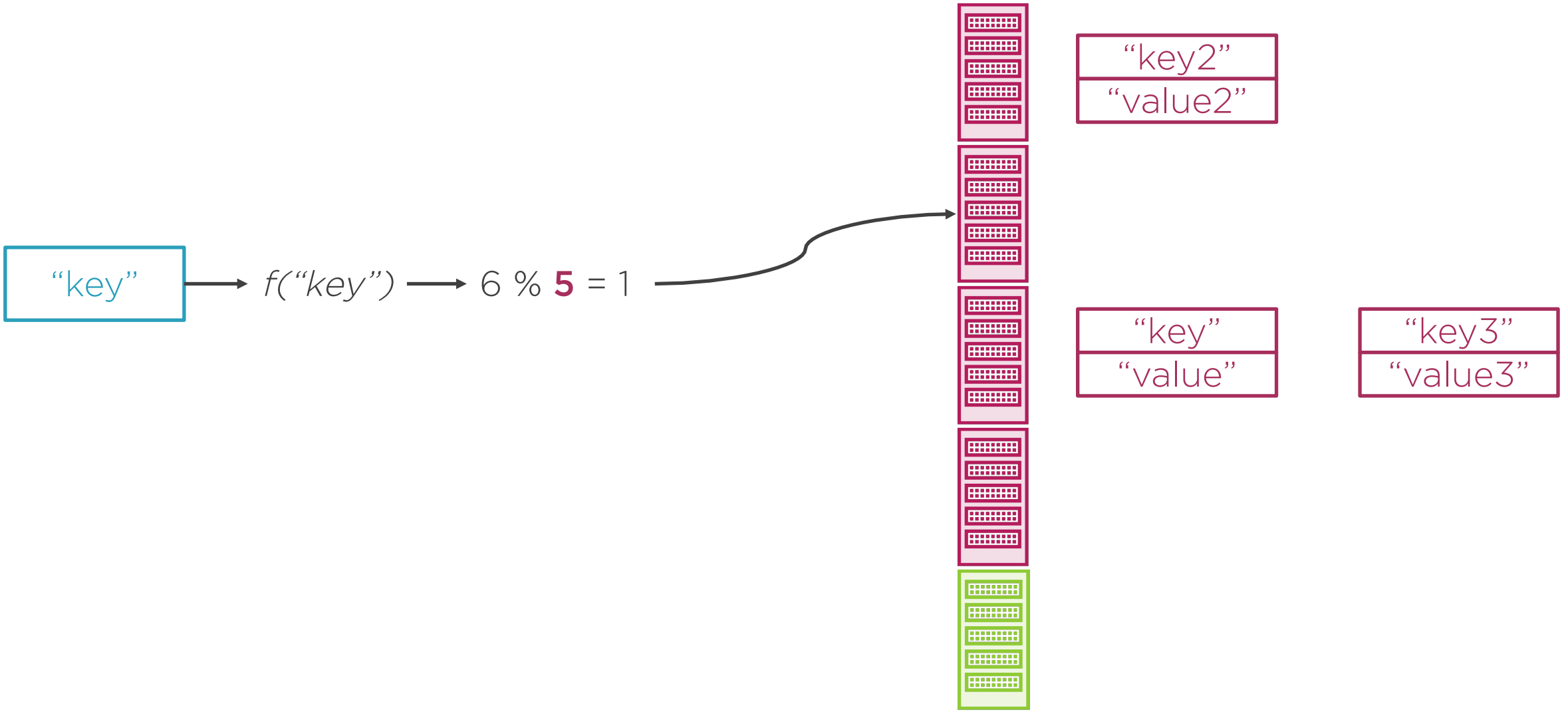
Hash Table Recap



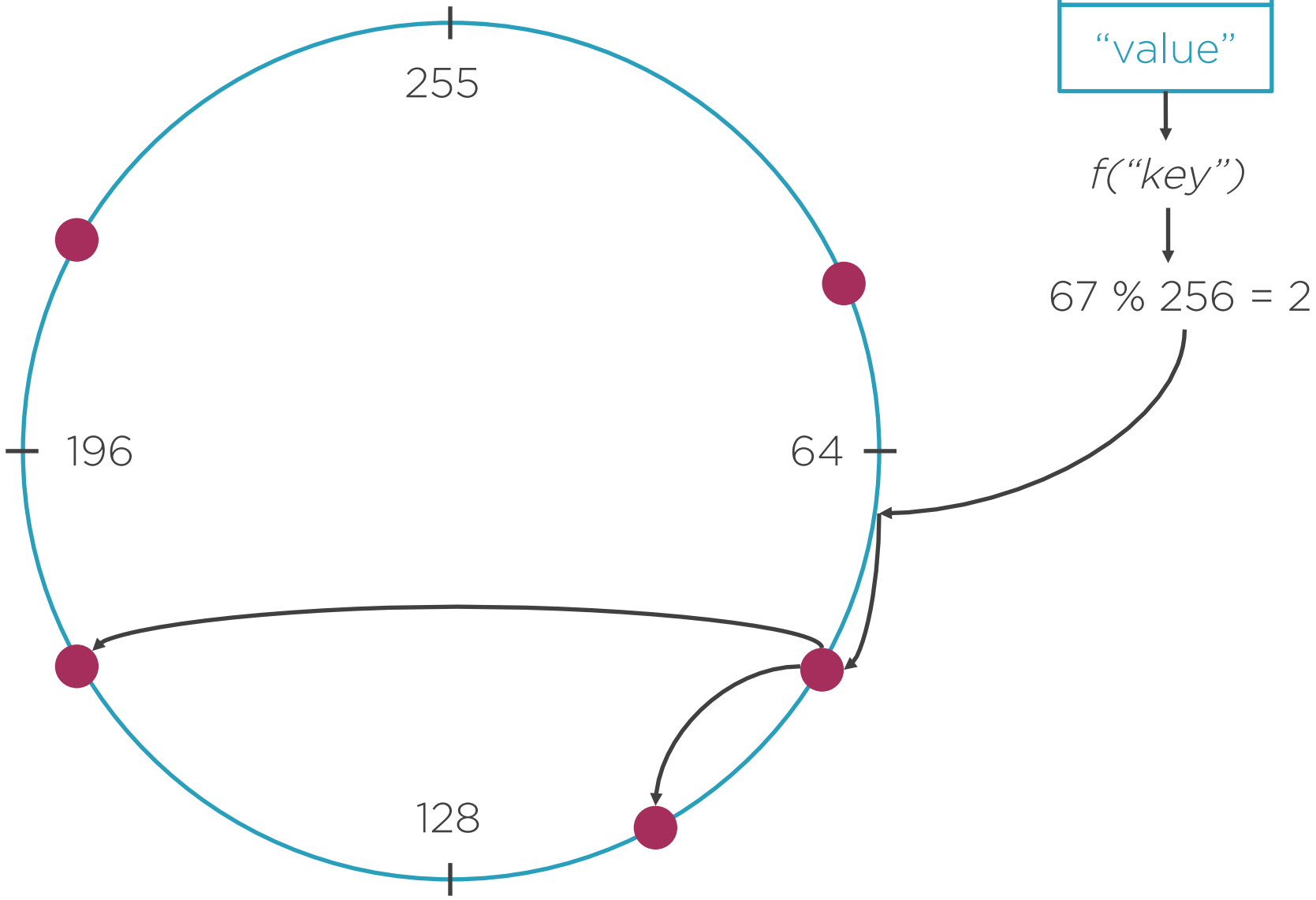
Hash Table Recap



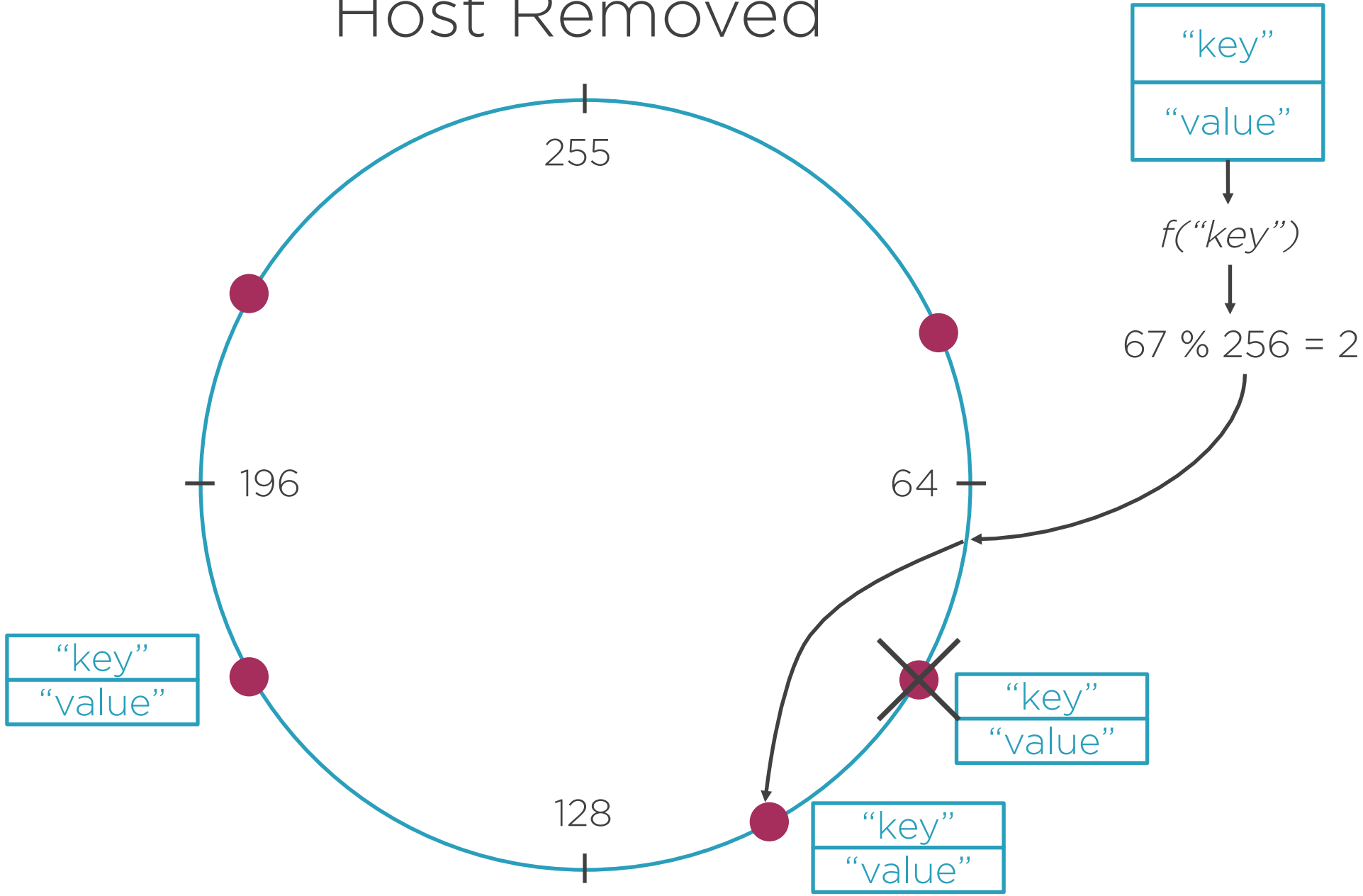
Hash Table with Servers



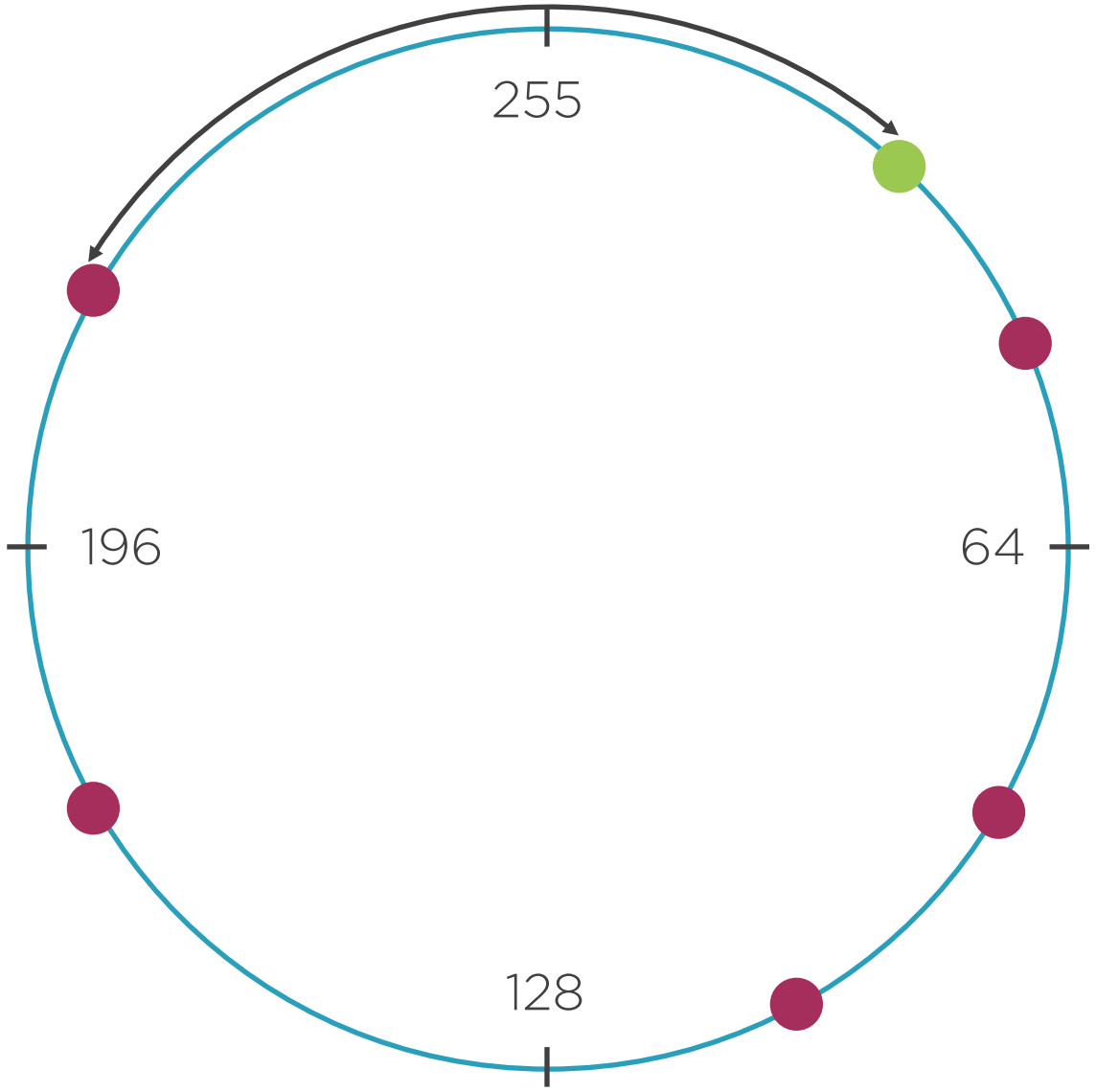
Consistent Hashing



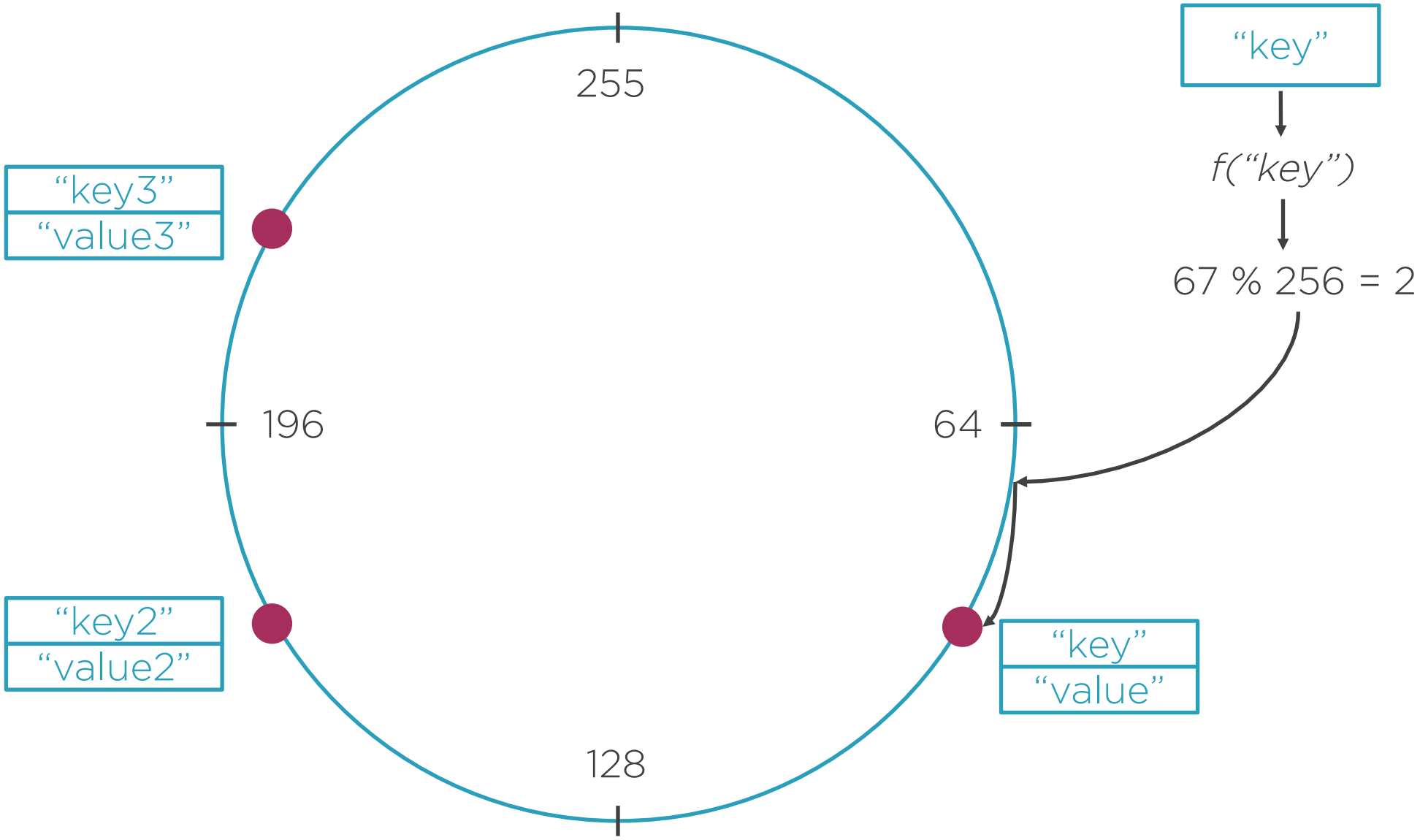
Host Removed



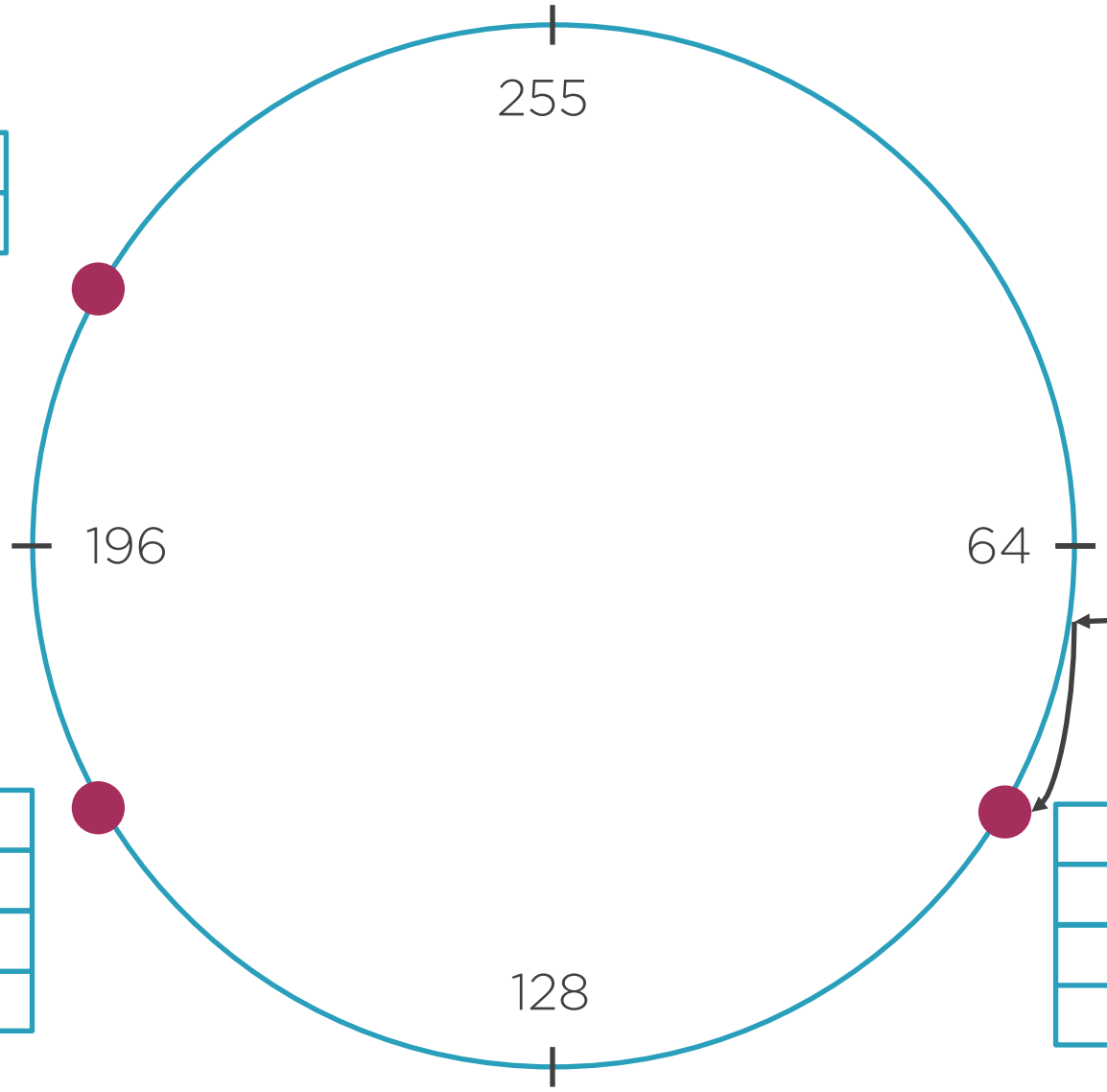
Host Added



Partition Keys



Partition and Sort Keys



UserId	Timestamp
4	100

UserId	Timestamp
5	400
5	600
5	725

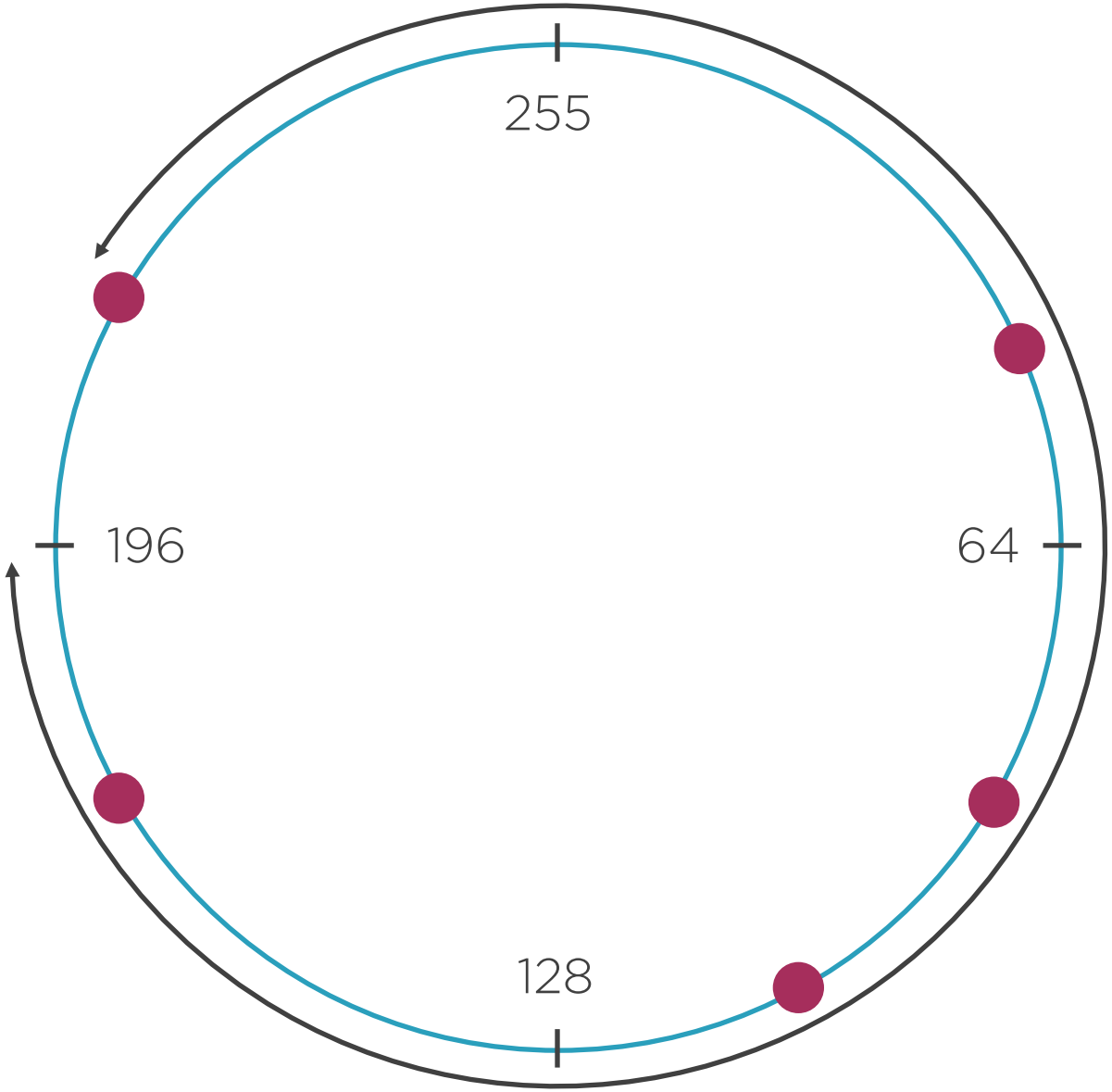
UserId
1

f(1)

67 % 256 = 2

UserId	Timestamp
1	100
1	200
2	125

Scan Execution



Partition Size Limit

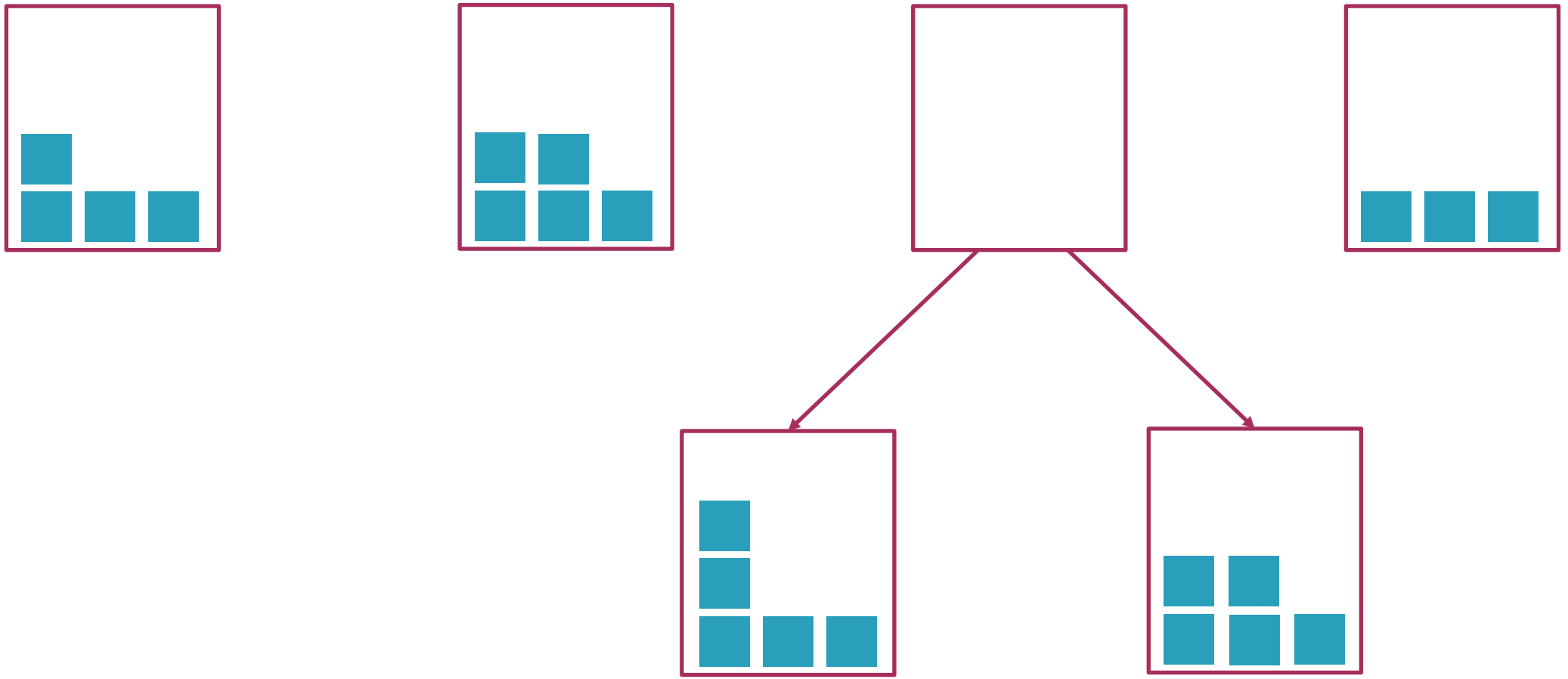
Partition can store up to 10GB

Items with different ids are moved to different partitions

Range key is limited to max size of a partition

Partition Split

Partition Split

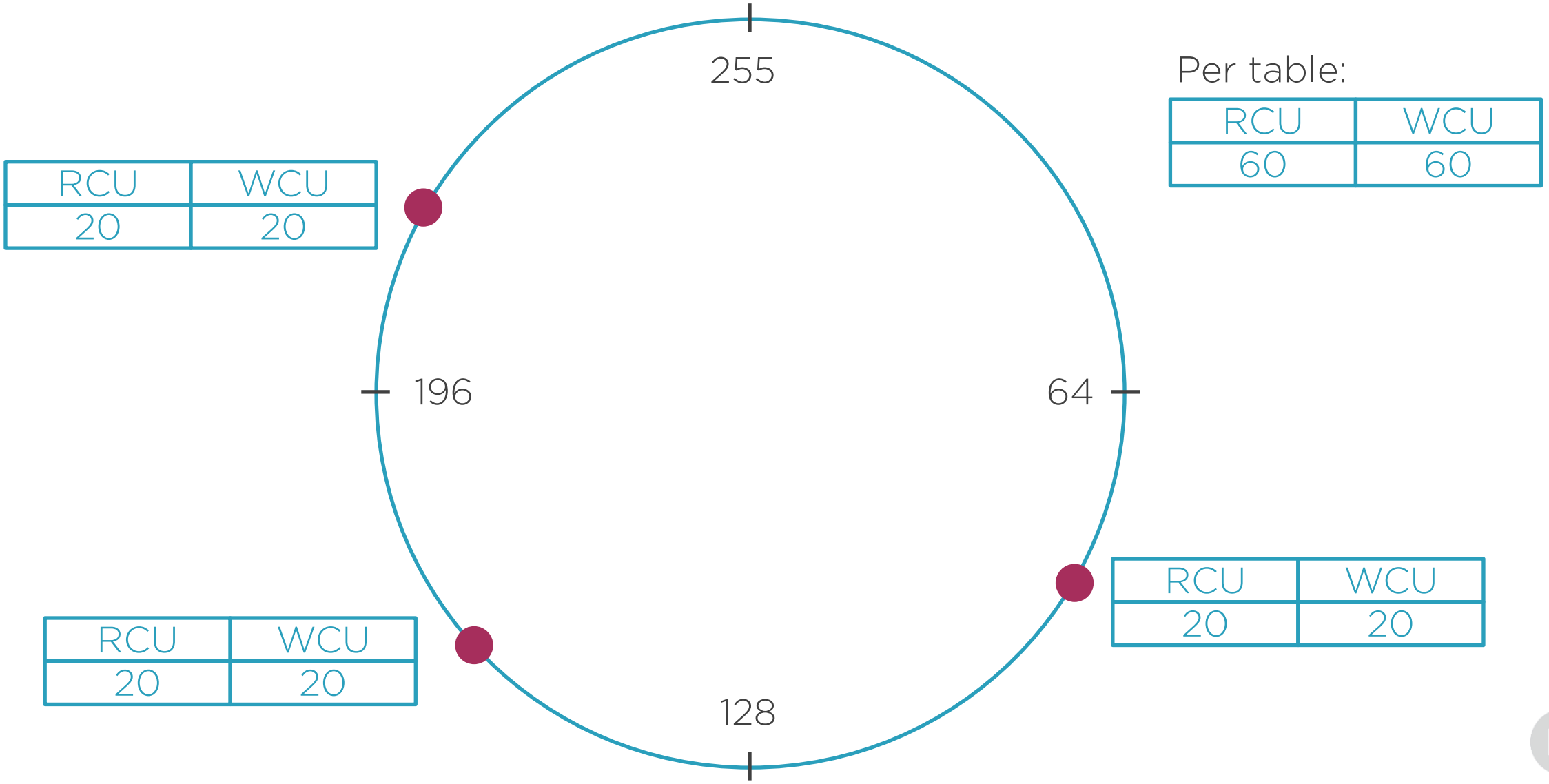


Number of Partitions

Number of partitions

By capacity	RCU / 3000 + WCU / 1000
By size	Total Size / 10GB
Result number	ceiling(max(capacity, size)

WCU/RCU Distribution



Databases Inspired by DynamoDB

Cassandra

Riak

Aerospike

Voldemort

Consistency in DynamoDB

Consistency in DynamoDB

Tradeoff of consistency

Data Consistency

Write updates all copies

Strongly consistent read – returns the most recent write

Eventually consistent read – may return some stale data

What Read Type to Use

Strongly consistent

A read requires at least 1 RCU

Should be used if you can't tolerate stale data

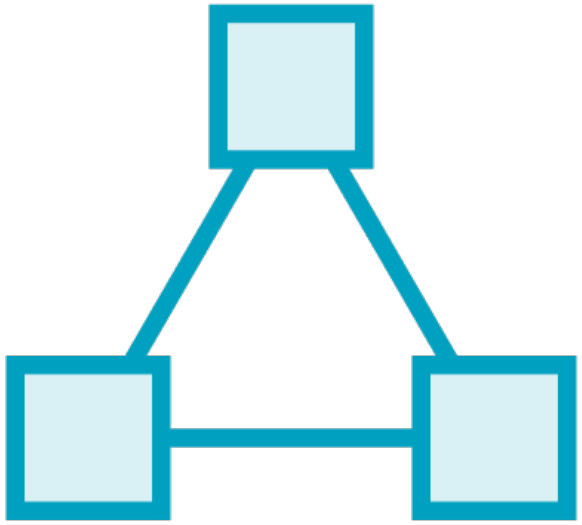
Eventually consistent

A read requires at least 0.5 RCU

Can be used when stale data is not an issue

Can help to reduce cost

Indexing Your Data



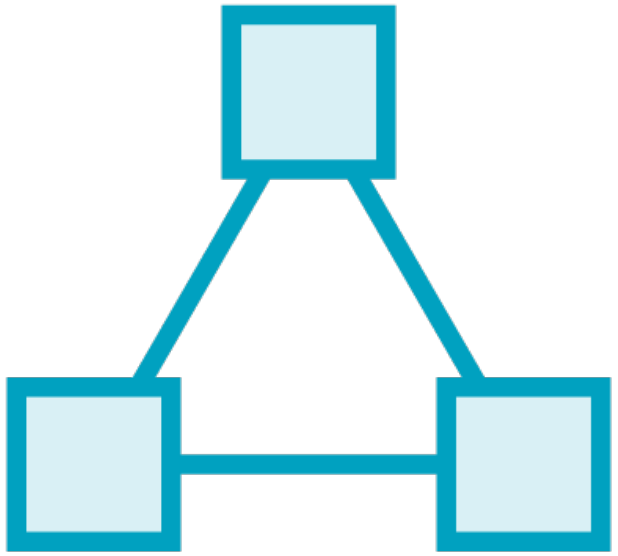
Indexes in DynamoDB

Why we need indexes

Types of indexes in DynamoDB

Indexes limitations

Why Do We Need Other Indexes



Primary keys are limiting

Queries they don't support:

- Sort posts from a user by rating or time
- Find all posts for topic/user

Data Example

UserId	Timestamp	Message	Rating	TopicId
1	1498916052	Lunch time?	5	1
1	1498915072	Just had my sandwich	4	1
1	1498928631	This forum is boring....	1	2
2	1498163954	Hello everybody	5	2
3	1497009531	Kittens photos here:	10	3

Keys types

Local secondary (LSI)

Select a different sort order for a partition key

Global secondary (GSI)

Access data using a different partition key

Local Secondary Indexes

Similar to sort keys

Values can be duplicated

Limited by size of a single partition

Find Top Posts by Author

Primary key

LSI

LSI

UserId	Timestamp	Message	Rating	TopicId
1	1498916052	Lunch time?	5	1
1	1498915072	Just had my sandwich	4	1
1	1498928631	This forum is boring....	1	2
2	1498163954	Hello everybody	5	2
3	1497009531	Kittens photos here:	10	3

Sort key

Global Secondary Indexes

- Copy with different partition key
- Always eventually consistent
- Unlimited size
- Can store subset of attributes
- You can emulate LSIs using GSIs

Find Top Posts by Author

Primary key

GSI Sort key

UserId	Timestamp	Message	Rating	TopicId
1	1498916052	Lunch time?	5	1
1	1498915072	Just had my sandwich	4	1
1	1498928631	This forum is boring....	1	2
2	1498163954	Hello everybody	5	2
3	1497009531	Kittens photos here:	10	3

Sort key

GSI Partition key

Indexes Limitations

Indexes have limitations

Up to 5 LSIs per table

Up to 5 GSIs per table

Single LSI can only be up to 10 GB (max partition size)

Demo

Create additional indexes

- Create local secondary index
- Create global secondary index

Query indexes

Pricing in DynamoDB

Pricing model

Learn what we are paying for

Avoid overpaying

Read Cost

Get one item

item-size / 4KB

Scan

size-of-all-read-items / 4KB

Range query

size-of-returned-items / 4KB

Get many items

sum of RCUs of individual items

Write Cost

Put item

item-size / 1KB

Update item

item-size / 1KB

Delete item

item-size / 1KB

Write many items

sum of WCUs of individual items

Additional Costs

Used storage

- First 25 GB stored per month is free
- $0.25 per GB-month thereafter

Data transfer

- All data in free
- Up to 1GB per month free
- Next 9,999TB per month $0.09 per GB
- Then progressively less per GB

Summary

- Basic DynamoDB concepts
- How to query data
- How to create indexes
- DynamoDB under the hood
- Learned how to use DynamoDB console
- Will learn DynamoDB API next