Persistence in AWS

Ryan Lewis
CLOUD ENGINEER
@ryanmurakami ryanlewis.dev

DynamoDB

Relational Database Service

ElastiCache

Overview

The details on DynamoDB

Bringing hamsters to the table

Starting a relationship with a database

Cluster the caches together

Persistent limits

How DynamoDB Provisioned Throughput Works

DynamoDB Throughput Capacity

The number of records that can be read or written per second. 4KB per unit for reading, 1KB per unit for writing.

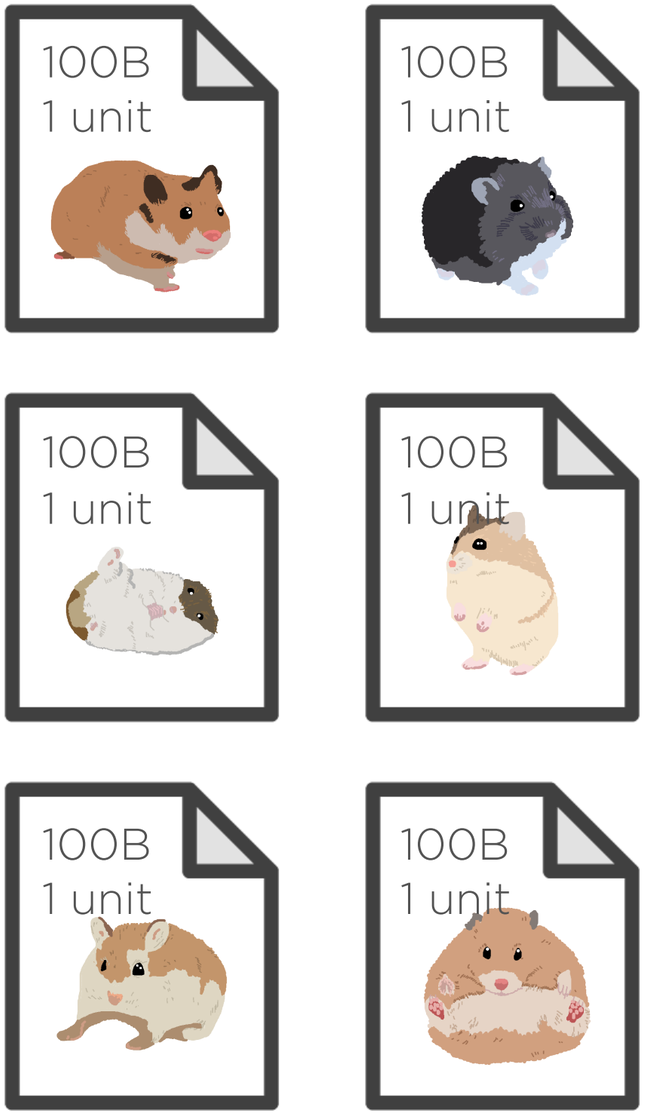
Hamsters Table

Provisioned Throughput

5 read units

5 write units

Batch Write



DynamoDB Burst Capacity

Used when throughput capacity is exceeded. No guarantees given from AWS of burst capacity availability.

Hamsters Table

Provisioned Throughput

5 read units

5 write units

Scan

6 records read

6 read units consumed

?

Eventually consistent reads
allow you twice the units

DynamoDB Read Types

Eventual consistency

May not have recent changes

Strong consistency

Guarantees newest changes

Hamsters Table

Provisioned Throughput

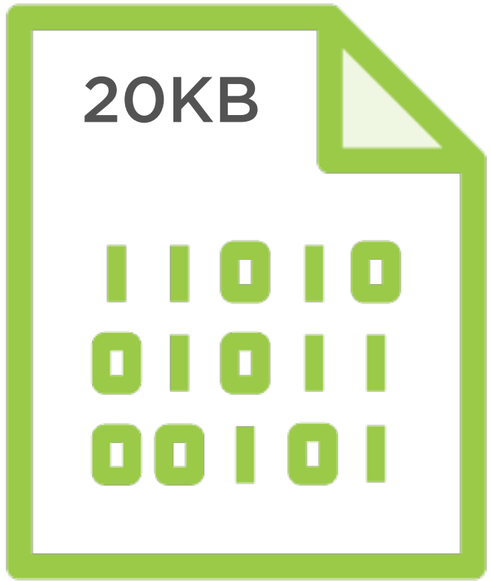
5 read units

5 write units

Scan

6 records read
with eventual consistency

3 read units consumed



Eventually Consistent Read - **3 units**

Strongly Consistent Read - **5 units**

Write - **20 units**

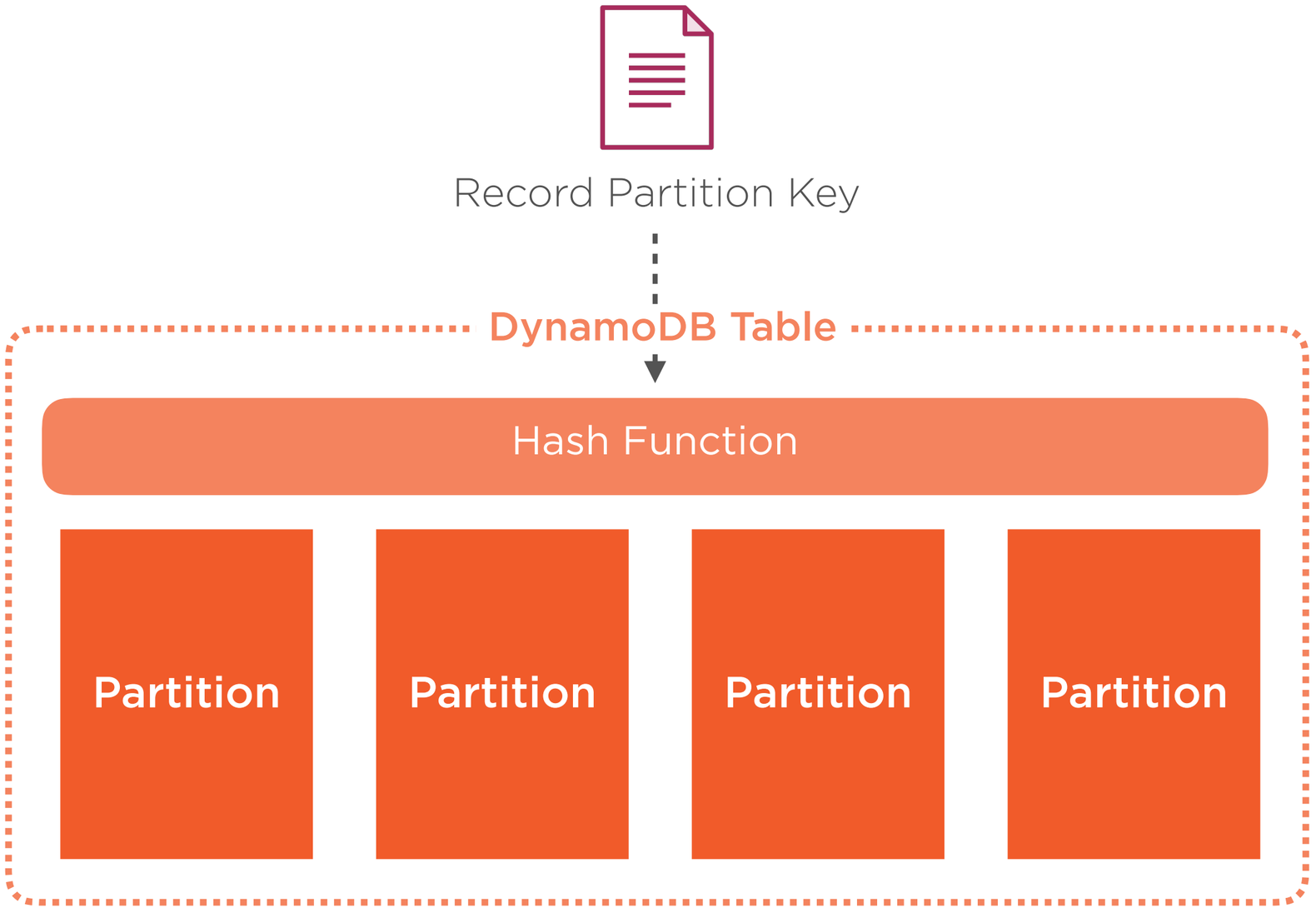
20KB / 8KB = 2.5 units
20KB / 1KB = 20 units
20KB / 4KB = 5 units

DynamoDB Keys and Secondary Indexes

Partition Key (Hash Attribute)

Used by a DynamoDB table to determine which partition to put a record. Must be unique if no range key used.

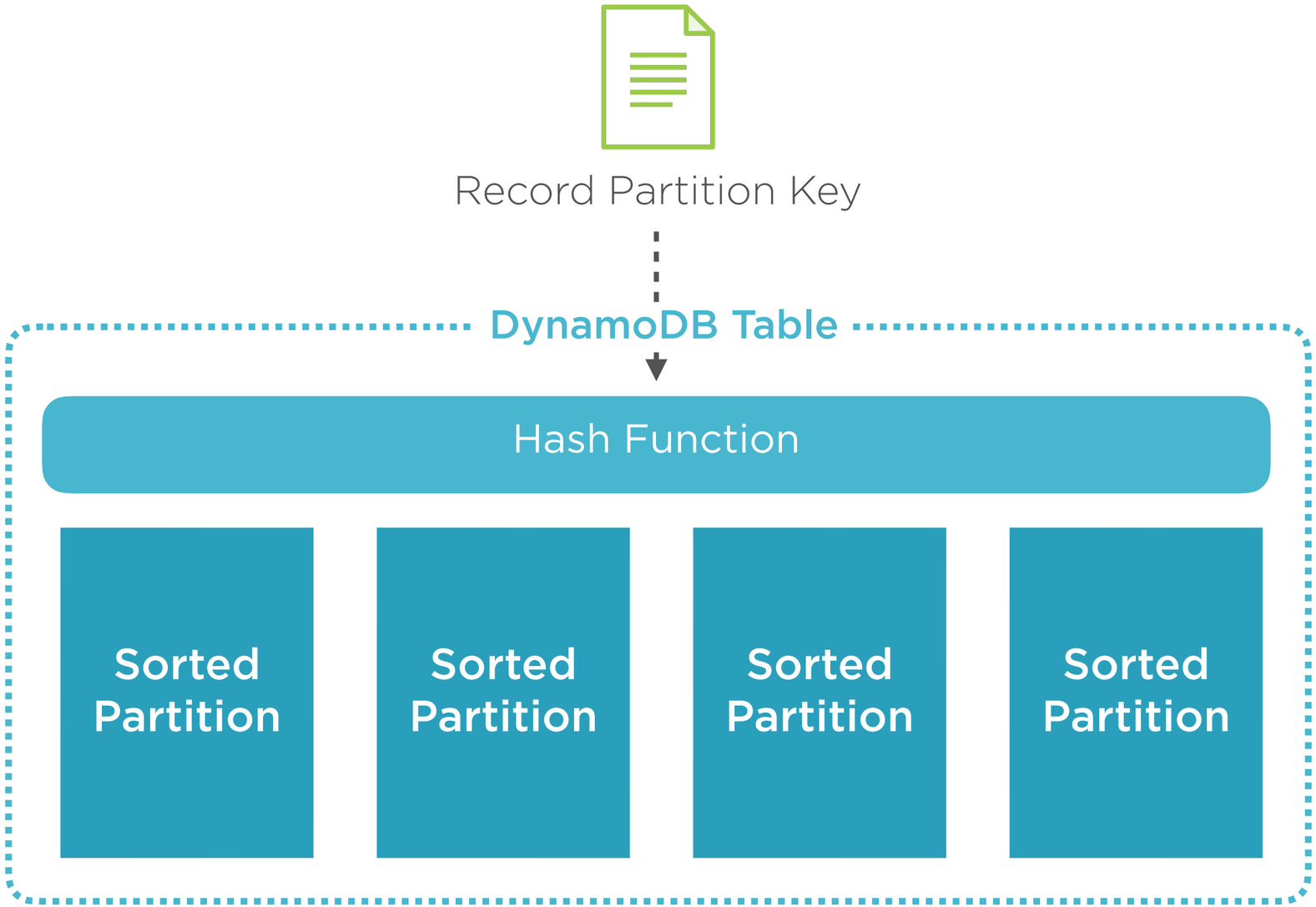
Partition Key Usage in DynamoDB



Sort Key (Range Attribute)

Used in conjunction with a partition key to sort documents with the same partition key in a partition.

Partition Key + Sort Key Usage in DynamoDB



DynamoDB Secondary Index Types

Global Secondary Index

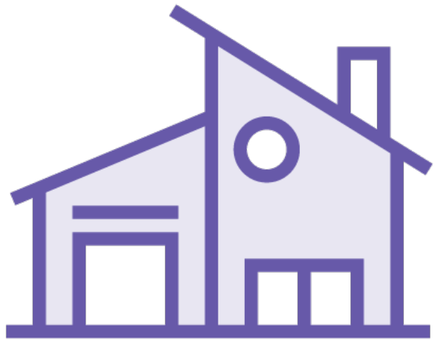
Local Secondary Index

Global Secondary Index

Define new key schema

Define record attributes to include in index

Independent provisioned throughput



Local Secondary Index

Define additional sort key only

Original partition key + new sort key used

All base table attributes available

Creating a DynamoDB Table

Populating a DynamoDB Table

Querying a DynamoDB Table

DynamoDB Retrieval Methods

Scan

Query

DynamoDB Table Scan

Retrieves all records from a table, 1MB at a time.

Creating a Database in RDS

Creating an ElastiCache Cluster

Limits with DynamoDB, RDS, and ElastiCache

DynamoDB Limit

5 global and local secondary indexes per table

DynamoDB Limit

Local secondary indexes must be created with the table

DynamoDB Limit

Only one table with secondary indexes can be created at a time

Relational Database Service Limit

Soft limits on number and size of databases

ElastiCache Limit

Soft limits on number of clusters and nodes

ElastiCache Limit

Clusters can’t be accessed outside of AWS

Conclusion

Summary

Indexing the provisioned throughput

Infesting DynamoDB with hamsters

Gotta catch ‘em all

MySQL for the users

Time for a Redis session

Elastic relational limits

Up Next

Routing from AWS