Creating, Querying and Modifying Our Data

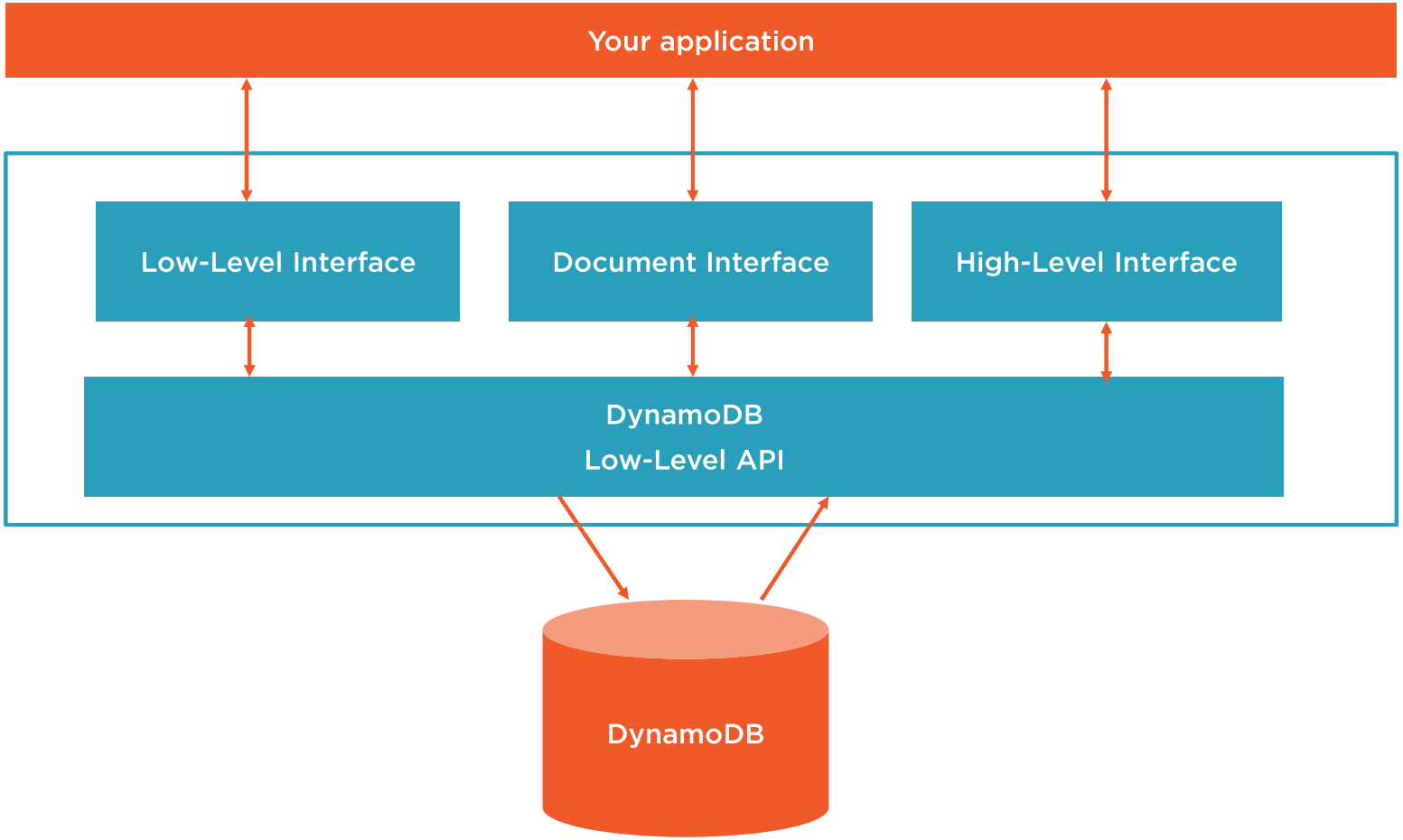
Abhaya Chauhan

@AbhayaChauhan www.abhayachauhan.com

Overview

- **AWS Management Console**
  - Create, Update and Delete items
- **AWS SDK for DynamoDB**
- **DynamoDB JSON Schema**
- **DynamoDB APIs**
  - Write, Read and Batch

AWS SDK


Your application
Low-Level Interface
Document Interface
High-Level Interface
DynamoDB
Low-Level API
DynamoDB

```
{
   "<AttributeName>":
     { "<DataType>": <AttributeValue> }
}
```

DynamoDB JSON Schema

DynamoDB uses a standard JSON format

```
{ "FirstName": { "S": "Abhaya" } }

{ "NoOfLogins": { "N": "0" } }

{ "Active": { "B": "true" } }
```

DynamoDB JSON - Scalar Data Types

{ "<*AttributeName*>": { "<*DataType*>": <*AttributeValue*> } }

```
{
  "SampleList": {
    "L": [ "Abhaya Chauhan", 5 ]
   }
}
```

DynamoDB JSON - List Data Types

{ "<*AttributeName*>": { "<*DataType*>": <*AttributeValue*> } }

```
{
  "StringSetExample": {
    "SS": [ "Abhaya", "Chauhan" ]
  }
}
```

DynamoDB JSON - List Data Types

{ "<*AttributeName*>": { "<*DataType*>": <*AttributeValue*> } }

```
{  "MapExample": {
     "M": {
       "FirstName": "Abhaya",
       "NoOfLogins": 0
} }
```

DynamoDB JSON - Map Data Types

{ "<*AttributeName*>": { "<*DataType*>": <*AttributeValue*> } }

DynamoDB's Write Operations

PutItem

UpdateItem

DeleteItem

Write API: PutItem

Creates a new item, or replaces an old item with a new item

If Item exists

- Item will be replaced

Otherwise

- Item will be created

Write API: UpdateItem

Edit an existing item's attributes

Or

Adds a new item to the table if it does not already exist

Write API: DeleteItem

Deletes a single item given a Primary Key

DynamoDB's Read Operations

GetItem

Query

Scan

Read API

Querying is eventually consistent by default

- For consistent reads, set ReadConsistent = true

All attributes available will be returned

- ProjectionExpressions can be used to limit attributes, but does not save RCUs

Read API: GetItem

Enables you to retrieve a single item given the Primary Key

Read API: GetItem

RCUs = Round up (Item size / 4KB) ÷ 2

Item Size = 6.2KB

Round up (6.2 / 4) ÷ 2
= 1 RCU

Read API: Query

Enables you to retrieve items based on primary key values

- Given Partition Key, and/or Sort Key
- Range of comparison operators available on the Sort Key

Read API: Query

RCUs = Round up (Item size / 4KB) ÷ 2

Item Size = 112KB

Round up (112 / 4) ÷ 2
= 14 RCUs

Read API: Scan

Returns one or more items by accessing every item in the table

Read API: Scan

RCUs = Round up (Item size / 4KB) ÷ 2

Item Size = 44KB

Round up (44 / 4) ÷ 2
= 5.5 RCUs (if we used Query API)

vs 127 RCUs (for the first page of Scan API)

DynamoDB's Batch Operations

BatchGetItem

BatchWriteItem

Batch API: BatchGetItem

Enables you to send one or more GetItem API calls in one request.

Batch API: BatchGetItem

Each operation can retrieve up to 16MB of data, or 100 items.

The batch call is not atomic

This batch operation works in parallel

Results are not ordered

Batch API: BatchWriteItem

Enables you to perform multiple puts or deletes in one request

Batch API: BatchWriteItem

- Up to 16MB of data, or up to 25 PUTs and DELETE requests
- Cannot update items
- Not atomic
- Works in Parallel
- Saves time, but not WCUs

Summary

DynamoDB JSON Schema

```
{
  "<AttributeName>":
    { "<DataType>": <AttributeValue> }
}
```

Summary

Write APIs

- PutItem
  - Create or Update entire item
- UpdateItem
  - Create new item or partial item update
    - Partial item updates save WCUs
- DeleteItem
  - Delete an existing item

Summary

Read APIs

- By Default, all Read API calls are eventually consistent

- GetItem
  - Get an item given the PK
- Query
  - Get items within a Partition Key
- Scan
  - Scans every item in the table
  - Avoid if possible – Very expensive

Summary

Batch APIs:

- BatchGetItem
  - One or more Get Item calls in one request
- BatchWriteItem
  - One or more PutItem or DeleteItem calls in one request

- Entire Batch API calls are not atomic