Leveraging DynamoDB in a Node.js Environment

Ryan Bickham
Sift

Detroit.Code()
July 12, 2017

Overview

- Introduction to DynamoDB
- Building a Node.js library to make things easier
- Serverless processing against DynamoDB

What is DynamoDB?

“Amazon DynamoDB is a fast and flexible NoSQL database service for all applications that need consistent, single-digit millisecond latency at any scale. It is a fully managed cloud database and supports both document and key-value store models.”

(https://aws.amazon.com/dynamodb/)

Data Schema

Table - A collection of items that share key attributes and indexes

Item - A document that contains one to many attributes, including a primary key

Attribute - A property on an item that has a type and a value

Primary Keys

Primary keys in Dynamo are made up of one or two attributes including a **partition key**, and an optional **sort key.**

The partition key influences how data is partitioned behind the scenes and can impact performance

Capacity and Scalability

- Each table has a provisioned read and write capacity, expressed in units. These units have no practical limit
- Units are closely related to item size

Indexes

Indexes allow a primary key like schema to be applied to different attributes.

Local secondary indexes are created using the partition key and an alternate sort key.

Global secondary indexes are created using an alternate key on any table attributes. These require new partitions to be created and stand-alone capacity.

Queries

A **query** in Dynamo requires at least a provided partition key, and can also include a sort key and/or additional filters.

A **scan** retrieves all items in the table without querying on a key.

Filters are additional conditions to match on for non-key attributes.

AWS Console Demo

Good

- “Infinitely” scalable
- Fully managed
- Quick key lookups
- Flexible (documents, filters)

Bad

- Can be expensive
- Finding the right capacity can be tricky. No built in scaling (until recently).
- APIs are tricky to use out of the box (at least in JavaScript)
- Vendor locked

DynamoDB SDK

- Requires a intermediate to advanced understanding of DynamoDB
- Request and response parameters are too complex to write code against quickly
- Requires knowledge of existing indexes when querying

Retrieving an Item

```
var params = {
 Key: {
  "Artist": {
    S: "Acme Band"
   },
  "SongTitle": {
    S: "Happy Day"
   }
 },
 TableName: "Music"
};
dynamodb.getItem(params, function(err, data) {
  if (err) console.log(err, err.stack);
  else     console.log(data);
});
```

Performing a Query

```
var params = {
 ExpressionAttributeValues: {
  ":v1": {
    S: "No One You Know"
   }
 },
 KeyConditionExpression: "Artist = :v1",
 TableName: "Music"
};
dynamodb.query(params, function(err, data) {
  if (err) console.log(err, err.stack);
  else     console.log(data);
});
```

A Better Way

- Any developer should be able to access the database easily
    - Simpler methods for reading and writing
    - Only a brief understanding of Dynamo required
- Items should have some kind of schema with flexible validation

Retrieving an item

```
dynamo.songs.getItem('Acme Band', 'Happy Day');
```

Performing a query

```
dynamo.songs.query({ Artist: 'No One You Know' });
```

Adding/Updating an item

```
let song = {
 Artist: 'No One You Know',
 SongTitle: ‘Nothing You Know'
};

dynamo.songs.putItem(song);
```

Deleting an Item

```
dynamo.songs.deleteItem(‘Acme Band', 'Happy Day');
```

Item Validation

- Code-based database schema
- Required attributes
- Attribute types
- Allowing or not allowing additional attributes

Library Code and Demo

DynamoDB Streams

- Capture changes to a table when an item is added, updated, or deleted
- Can be subscribed to via an API endpoint or through many AWS services, including Lambda

AWS Lambda

- Run code without managing servers, and pay only for use
- Triggered by AWS services including Dynamo, S3, API Gateway, and more

Streams and Lambda Demo

Thank You!

- Slides/code at github.com/rbickham11/DynamoNodeSlides
- Email: ryan@justsift.com

We’re Hiring!

justsift.com/jobs

Questions?