

COSC 416A: Special Topics in Databases... DynamoDB

John Pick, Drew Ingram and Rickie Tsang

What is DynamoDB?

- DynamoDB is a database service offered by Amazon
- It uses Key-Value storage
- It is designed to provide predictable performance and take the hassle out of scaling your database for larger volumes of data.

The back end

- Data is stored on Solid-state drives for low cost and fast performance
- Data is automatically replicated across AWS availability zones
- Scaling is handled automatically – You don't have to have the time or expertise to maintain your database when there is a large increase in the number of users

Data representation

- Attributes are individual pieces of data (ex: name)
- Attributes can have values of numbers, strings, binary, number sets, string sets or binary sets.
- Items are made up of several attributes (ex: name = “Bob”, rating = “lame”).
- Items must have a primary key.
- Primary keys may consist of a single hash value, or a hash value and a range value.
- Tables are made up of several items.

Primary keys

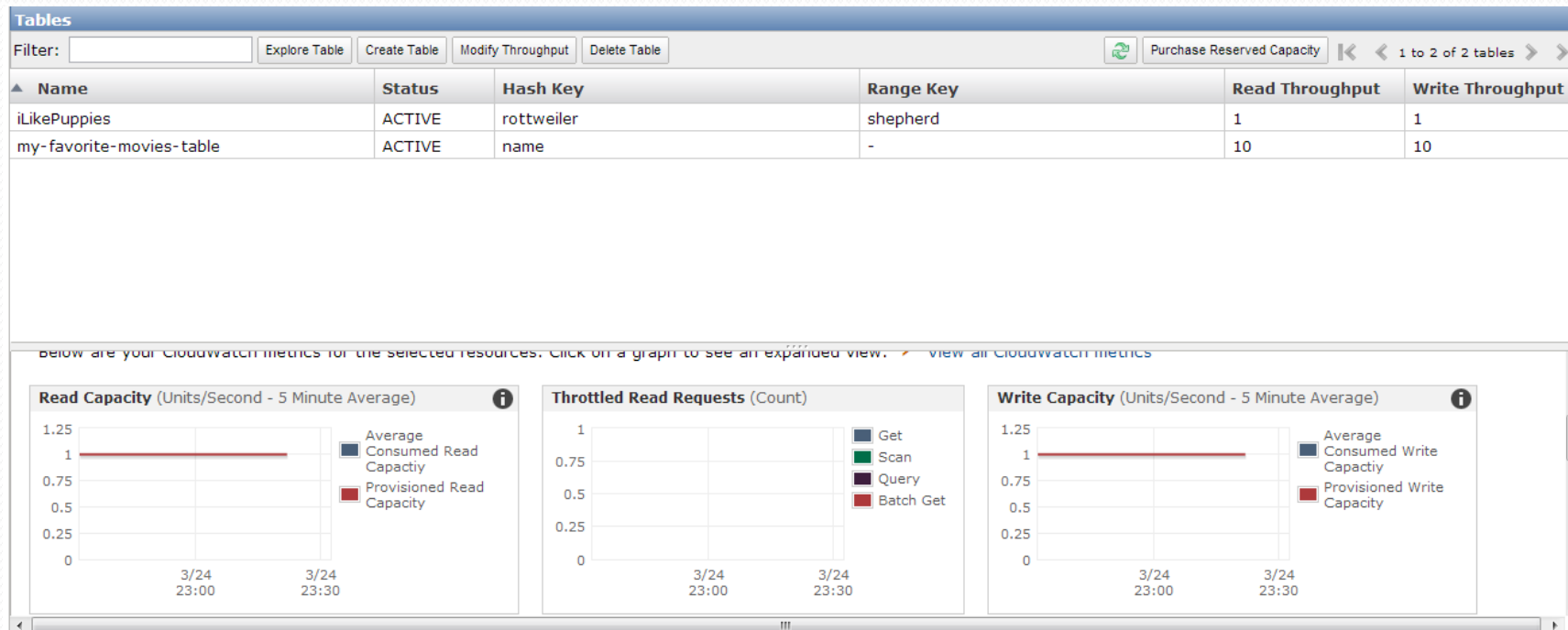
- Primary keys consist of either a hash key, or a hash key AND a range key
- The hash key is used to uniquely identify the item
- The optional range key is used for sorting. For example, a table containing games might have a range key “Release Date” that you could use to sort the games from earliest to latest release date.
- The hash key for two different items can be the same, as can the range key, AS LONG AS they hash key/range key combination is unique

DynamoDB schema

- DynamoDB does not use a schema
- The primary key attributes are the only ones required for a table in DynamoDB. Any other attribute are entirely optional

The AWS console

- Data can be managed from the AWS management console
- From here, you can add and removes tables and data from those tables
- When you create a table, you must wait while it becomes active



AWS console continued...

- Manage items

Name	Number	Points	Trophies	Save Percentage	Tropies
"Kevin Bieksa"	3	7			
"Cory Schneider"	35		{ "Jennings" }	0.92	
"Henrik Sedin"	33	27			{ "Art Ross", "Hart" }
"Daniel Sedin"	22	"28"	{ "Art Ross", "Ted Lindsey" }		

- The titles across the top are the attribute names
- The first attribute “Rottweiler” is the hash key
- The second, “Shepherd” is the range key
- The others are attributes unique to each item. If an item does not have an attribute in another item, the field is left blank

Querying and adding data without the console

- DynamoDB does not have its own query language (no putty)
- Instead, it supports several different languages including:
- Java
- JavaScript
- Mobile
- PHP
- Python
- Ruby
- Windows & .NET

Creating Table

N - Number

S - String

SS - String Set

```
String tableName = "TableName";
```

```
CreateTableRequest createTableRequest = new  
    CreateTableRequest().withTableName(tableName).withKeySchema(new  
    KeySchema(newKeySchemaElement().withAttributeName(AttributeName).withAttribute  
    eType("N")).withProvisionedThroughput(newProvisionedThroughput().withReadCap  
    acityUnits(1L).withWriteCapacityUnits(1L));
```

```
waitForTableToBecomeAvailable(tableName);
```

Adding elements to table

```
String tableName = "TableName";
Map<String, AttributeValue> item = new HashMap<String, AttributeValue>();
item.put("name", new AttributeValue(name));
item.put("year", new AttributeValue().withN(Integer.toString(year)));
item.put("fans", new AttributeValue().withSS(fans));
PutItemRequest putItemRequest = new PutItemRequest(tableName, item);
PutItemResult putItemResult = dynamoDB.putItem(putItemRequest);
```

Scan item / Scan item with filter

Get items from DB

```
ScanRequest scanRequest = new ScanRequest(tableName);
ScanResult scanResult = dynamoDB.scan(scanRequest);
```

Get item from DB with filter (year > 1985)

```
HashMap<String, Condition> scanFilter = new HashMap<String, Condition>();
Condition condition = new Condition()
    .withComparisonOperator(ComparisonOperator.GT.toString())
    .withAttributeValueList(new AttributeValue().withN("1985"));
scanFilter.put("year", condition);
ScanRequest scanRequest = new
    ScanRequest(tableName).withScanFilter(scanFilter);
ScanResult scanResult = dynamoDB.scan(scanRequest);
```

Free Tier

- Because DynamoDB is a service, it costs money
- You have a certain “Free Tier” of 100MB of storage space, 5 writes per second and 10 reads per second
- You must ensure that you do not exceed this free tier to avoid incurring a small charge on your credit card

Pros and Cons

- *The good...*
- The fast, predictable performance
- No need to worry about scaling
- *And the bad...*
- Cannot do complex queries (No joins, no Group By, no aggregate functions, can't sort on anything apart from the range key)

Some useful references...

- <http://aws.amazon.com/dynamodb/> DynamoDB's homepage
- <http://aws.amazon.com/documentation/dynamodb/> Dynamo's documentation
- <http://www.youtube.com/watch?v=yqlauXAXlSc> The first video in a playlist of four about building applications using DynamoDB
- <http://www.youtube.com/watch?v=oz-7wJJ9HZo> An overview of DynamoDB